INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization*

Régis Duvigneau

**N° 5949**

Juillet 2006

Thème NUM


ISSN 0249-6399 ISRN INRIA/RR--5949--FR+ENG

# Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization

Régis Duvigneau*

Thème NUM — Systèmes numériques
Projet Opale

Rapport de recherche n° 5949 — Juillet 2006 — 40 pages

**Abstract:** Parameterization techniques commonly used in aerodynamic shape optimization are essentially general and multi-purpose approaches. As a consequence, they cannot be well suited to a particular shape optimization problem. The present study proposes a new method that adapts an initial and perhaps naïve parameterization of an aerodynamic shape by the Free-Form Deformation (FFD) technique, to the particular optimization problem to solve, according to a first approximation of the solution. This parameterization adaption method is included in a three-dimensional aerodynamic shape optimization procedure for Eulerian flows and is demonstrated for the design optimization of the wing of a business aircraft.

**Key-words:** Parameterization, Free-Form Deformation, Adaption, Shape optimization, Computational Fluid Dynamics, Aerodynamics

* Opale Project-Team

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

# Paramétrisation adaptative par la méthode des boîtes englobantes pour l'optimisation de forme aérodynamique

**Résumé :** Les techniques de paramétrisation utilisées habituellement en optimisation de forme aérodynamique sont par essence générales et multiobjectives. Par conséquent, elles ne peuvent pas être bien adaptées à un problème d'optimisation de forme en particulier. Cette étude propose une nouvelle méthode qui adapte une paramétrisation initiale et peut-être naïve d'une forme aérodynamique par la méthode des boites englobantes, au problème d'optimisation particulier que l'on souhaite résoudre, sur la base d'une première approximation de la solution. Cette méthode d'adaptation de la paramétrisation est inclue dans une procédure d'optimisation de forme aérodynamique tridimensionnelle pour des écoulements Euleriens et mise en oeuvre pour l'optimisation de la forme de l'aile d'un jet d'affaire.

**Mots-clés :** Parametrisation, Boîtes englobantes, Adaptation, Optimization de forme, Mécanique des Fluides Numérique, Aérodynamique

# Contents

# Introduction

Optimum shape design in aerodynamics has been an active research topic for several years. The recent improvements of numerical methods in Computational Fluid Dynamics (CFD) and in optimization, as well as the developement of high performance computing capabilities, yield a significant reduction of the design time cycle. Therefore, the three-dimensional aerodynamic shape optimization of parts of an aircraft is today a common exercise in aeronautics.

A critical issue in shape optimization is the choice of the shape parameterization. The objective of the parameterization is to describe the shape, or the shape modification, by a set of parameters which are considered as design variables during the optimization procedure. Parameterization techniques in shape optimization have to fulfill several practical criteria :

- the parameterization should be able to take into account complex geometries, possibly including constraints and singularities ;
- the number of parameters should be as small as possible, since the stiffness of the shape optimization numerical formulation increases abruptly with the number of parameters ;
- the parameterization should allow to control the smoothness of the resulting shapes ;
- the parameterization should be consistent with grid update approaches and with domain partitionning methods.

A survey of shape parameterization techniques, which are analyzed according to the previous criteria, is proposed by Samareh [10, 11]. In accordance with his conlusions, the Free-Form Deformation (FFD) technique [12] is adopted in the present study, since it provides an easy and powerful framework for the deformation of complex shapes, as those encountered in aerodynamics.

Whatever the parameterization method chosen by the user is, the main issue in shape parameterization for optimization purpose is the critical dependency of the optimum shape found on the parameterization. Indeed, the choice of the parameterization determines the set of shapes that can be reached during the optimization procedure. Since the optimum shape found belongs to this set, it depends obviously on the parameterization. As a consequence, the choice of the parameterization influences the results of the optimization, in terms of accuracy of the solution (i.e. the distance between the optimum shape found and the solution) and convergence rate (related to the stiffness of the problem).

The present study proposes a method to automatically adapt an initial and perhaps naïve parameterization of an aerodynamic shape by the FFD technique, to the particular optimization problem to solve, according to a first approximation of the solution. The proposed method is inspired from prior works based on Bézier curves and applied to a more simple geometrical two-dimensional problem [7]. It was found [9] that the regularization of the control polygon of the Bézier curve yields a significant improvement in terms of accuracy. The present work aims at extending these developements to 3D aerodynamic shape optimization.

The parameterization adaption method is described in the first section. Then, the two-dimensional parameterization of an airfoil is used to illustrate the proposed approach. These developements are then considered in the framework of aerodynamic shape optimization. After the baseline of the procedure is reminded (Eulerian flow solver, grid update, optimizer), the adaption method is applied to the aerodynamic shape optimization of the wing of a business aircraft.

# 1 Free-Form-Deformation parameterization

The Free-Form-Deformation (FFD) technique originates from the Computer Graphics field [12]. It allows the deformation of an object in a 2D or 3D space, regardless of the representation of this object. Instead of manipulating the surface of the object directly, by using classical B-Splines or Bézier parameterization of the surface, FFD defines a deformation field over the space embedded in a lattice which is built around the object. By transforming the space coordinates inside the lattice, FFD deforms the object, regardless of its geometrical description.

More precisely, consider a three-dimensional hexaedral lattice embedding the object to be deformed. Figure (1) shows an example of such a lattice built around a realistic wing. A local coordinate system $(\xi, \eta, \zeta)$ is defined in the lattice, with $(\xi, \eta, \zeta) \in [0,1] \times [0,1] \times [0,1]$. During the deformation, the displacement $\Delta q$ of each point $q$ inside the lattice is here defined by a third-order Bézier tensor product :

$$\Delta q = \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} B_i^{n_i}(s_q) B_j^{n_j}(t_q) B_k^{n_k}(u_q) \Delta P_{ijk}. \tag{1}$$

$B_i^{n_i}$, $B_j^{n_j}$ and $B_k^{n_k}$ are the Bernstein polynomials of order $n_i$, $n_j$ and $n_k$ (see for instance [8]) :

$$B_p^n(t) = C_n^p \, t^p \, (1-t)^{n-p}. \tag{2}$$

In (1), $(s_q, t_q, u_q)$ are the FFD coordinates of the point $q$ which result from a mapping of the lattice coordinates $(\xi_q, \eta_q, \zeta_q)$. Usually this mapping is simply defined as the identity. But, as will be explained latter, there might be some advantages of using an other mapping. Finally, $\Delta P_{ijk}$ are weighting coefficients, or control points displacements, which are used to monitor the deformation and are considered as design variables during the shape optimization procedure.

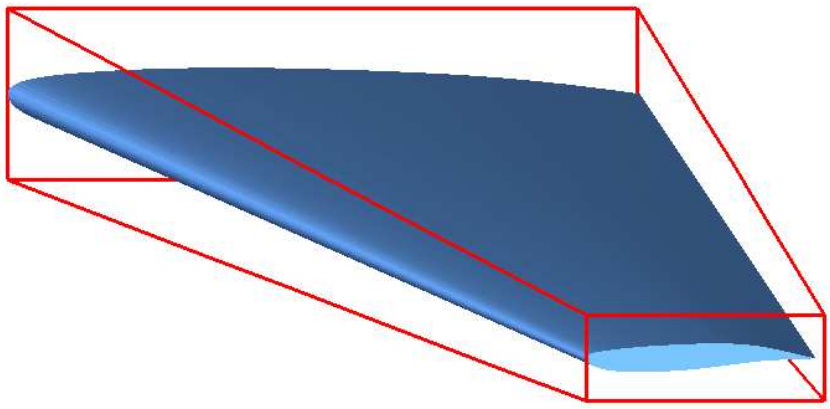


Figure 1: Example of FFD lattice (red) around a wing.

The FFD deformation technique described above is well suited to complex shape optimization, thanks to the following properties :

- the initial shape can be exactly represented (no deformation occurs when all weighting coefficients are zero) ;
- the deformation is performed whatever the complexity of the shape (this is a free-form technique) ;
- geometric singularities can be taken into account (the initial shape including its singularities is deformed) ;
- the smoothness of the deformation is controled (the deformation is ruled by Bernstein polynomials) ;
- the number of design variables depends on the user's choice (the deformation is independent of the shape itself) ;
- it nicely deals with multi-level representation (thanks to the Bézier degree elevation property).

A new notation is introduced to facilitate further developements. Consider the vectors :

$$\begin{aligned} B^T =& [B_i^{n_i}(s_q) B_j^{n_j}(t_q) B_k^{n_k}(u_q)]_{i=1,\ldots,n_i\,;\,j=1,\ldots,n_j\,;\,k=1,\ldots,n_k} \in \Re^{(n_i+1)\times(n_j+1)\times(n_k+1)} \\ =& [B_0^{n_i}(s_q) B_0^{n_j}(t_q) B_0^{n_k}(u_q), B_0^{n_i}(s_q) B_0^{n_j}(t_q) B_1^{n_k}, \ldots, B_{n_i}^{n_i}(s_q) B_{n_j}^{n_j}(t_q) B_{n_k}^{n_k}(u_q)], \end{aligned} \tag{3}$$

$$\begin{aligned} \Delta P^T =& [\Delta P_{ijk}]_{i=1,\ldots,n_i\,;\,j=1,\ldots,n_j\,;\,k=1,\ldots,n_k} \in \Re^{(n_i+1)\times(n_j+1)\times(n_k+1)} \\ =& [\Delta P_{000}, \Delta P_{001}, \Delta P_{002}, \ldots, \Delta P_{n_i n_j n_k}]. \end{aligned} \tag{4}$$

Then, the third-order tensor product (1) can be expressed as a simple inner product on $\Re^{(n_i+1)\times(n_j+1)\times(n_k+1)}$ :

$$\Delta q = B^T \Delta P. \tag{5}$$

These notations are used in the next section to improve the legibility of the formulas.

# 2 Parameterization adaption procedure

## 2.1 Framework

The FFD parameterization presented above allows an easy modification of the shape of an object by modifying the weighting coefficients $\Delta P_{ijk}$ during the optimization procedure. However, this representation is not "universal", i.e. the reachable shapes depend on some choices of the user. For instance, if the mapping producing the FFD coordinates $(s,t,u)$ is modified, or if B-Splines basis functions are employed instead of Bernstein polynomials, or if basis functions of higher degree are used, then the reachable shapes will not be the same ones. Actually, the choice of the parameterization defines the mathematical subspace (of finite dimension) in which optimization occurs. If a poor choice is made by the user, the best reachable shape will be far from the solution (of infinite dimension) and the cost function will become more complex, yielding a stiffer optimization problem.

The aim of the present work is to develop a methodology to adapt an initial and perhaps naïve parameterization to the particular problem studied, reducing so the influence of the initial choice of the user and *increasing the robustness* of the overall procedure. In the framework of aerodynamic shape optimization, two outcomes are expected :

- reach a shape of better fitness (i.e. decrease the distance between the optimum shape found and the solution) ;
- increase the convergence rate (i.e. improve the conditioning of the optimization problem).

## 2.2 Principles

A direct approach to monitor adaption and modify the FFD parameterization presented above is to use the mapping producing the FFD coordinates $(s,t,u)$ from the lattice coordinates $(\xi,\eta,\zeta)$. Indeed, *this mapping controls the location of the points that are the mostly influenced by the design variables* in the FFD formulation (1). By modifying this mapping, one can control the location of the deformation that is obtained by perturbating each weighting coefficient. Therefore, the modification of the mapping seems to be a good means to control the parameterization.

The adaption procedure occurs at some steps of the shape optimization procedure. It is not clear at the present time when the adaption should be carried out optimally. The shape has to be close enough to the optimum shape in order to provide suitable a information to adapt the parameterization. However, one would like adaption to occur before the full convergence of the optimization is achieved, in order to improve the convergence rate. Therefore, further experiments will be required to determine a suitable criterion to perform adaption.

The adaption procedure consists in minimizing an adaption cost function (which measures the uneffectiveness of the current parameterization) by modifying some adaption parameters, which define the mapping. However, the adaption process should let the current shape unaltered, at least in a least-squares sense, in order to benefit from the optimization path already performed. Then, the minimization of the adaption cost function should be subject to the constraint that the resulting shape fits the current shape in a least-square sense. Whatever the nature of the adaption cost function and parameters, the adaptive shape optimization procedure is finally organized as follows :

1. Initialization
   Choose an initial shape $S^{(0)}_{\mathcal{M}_0}$ corresponding to the initial design variables $\Delta P^{(0)}_{\mathcal{M}_0} = 0$ for the initial mapping $\mathcal{M}_0$ (identity for instance).
2. Beginning of the loop over adaption steps
   set $l \leftarrow 0$
3. Shape optimization
   Perform $k_{opt}$ optimization steps yielding the shape $S^{(k_{opt})}_{\mathcal{M}_l}$ defined by the design variables $\Delta P^{(k_{opt})}_{\mathcal{M}_l}$.
4. Parameterization adaption
   Update the mapping to $\mathcal{M}_{l+1}$ by minimizing the adaption cost function, subject to the constraint that the shape $S^{(0)}_{\mathcal{M}_{l+1}}$ resulting from the mapping change and defined by the design variables $\Delta P^{(0)}_{\mathcal{M}_{l+1}}$ is the least-squares approximation of $S^{(k_{opt})}_{\mathcal{M}_l}$.
5. End of the loop
   set $l \leftarrow l+1$
   If stopping criterion reached than stop.
   Else Goto 2.

Before the adaption cost function and parameters are defined, the least-squares approximation in the framework of the FFD deformation is described in the next section.

## 2.3 Least-squares approximation

When the current mapping $\mathcal{M}_l$ is updated to a given mapping $\mathcal{M}_{l+1}$, the shape should remain as unaltered as possible to benefit from the optimization path already performed. Suppose that the current shape $S^{(k_{opt})}_{\mathcal{M}_l}$ is defined by the design variables $\Delta P^{(k_{opt})}_{\mathcal{M}_l}$. Then, the shape $S^{(0)}_{\mathcal{M}_{l+1}}$ resulting from the mapping change should be the least-squares approximation of $S^{(k_{opt})}_{\mathcal{M}_l}$. This approximation is obtained by considering the FFD deformation defined by the design variables $\Delta P^{(0)}_{\mathcal{M}_{l+1}}$ which minimizes the gap between the current shape $S^{(k_{opt})}_{\mathcal{M}_l}$ for the mapping $\mathcal{M}_l$ and the new shape $S^{(0)}_{\mathcal{M}_{l+1}}$ for the mapping $\mathcal{M}_{l+1}$ :

$$\text{Minimize} \iint_{S^{(0)}_{\mathcal{M}_0}} \frac{1}{2}[\Delta q^{(0)}_{\mathcal{M}_{l+1}} - \Delta q^{(k_{opt})}_{\mathcal{M}_l}]^2 dS, \tag{6}$$

where $\Delta q^{(0)}_{\mathcal{M}_l}$ and $\Delta q^{(k_{opt})}_{\mathcal{M}_{l+1}}$ are the displacements of a point $q$ from the initial surface $S^{(0)}_{\mathcal{M}_0}$, for the FFD deformations producing respectively the current shape (mapping $\mathcal{M}_l$) and the new approximated shape (mapping $\mathcal{M}_{l+1}$). However, this integral cannot be evaluated in the FFD framework since no curvilinear description of $S^{(0)}_{\mathcal{M}_0}$ is available. Therefore, the discrete form of the least-squares approximation is preferred, by minimizing :

$$\mathcal{J}_{LS} = \sum_{n=1}^{N} \frac{1}{2}[\Delta^n q^{(0)}_{\mathcal{M}_{l+1}} - \Delta^n q^{(k_{opt})}_{\mathcal{M}_l}]^2 \delta S^n, \tag{7}$$

where $\Delta^n q^{(0)}_{\mathcal{M}_{l+1}}$ and $\Delta^n q^{(k_{opt})}_{\mathcal{M}_l}$ represent the displacements of the mesh node $q^n$ from the initial surface $S^{(0)}_0$ for the FFD deformations producing respectively the new shape and the current shape. $\delta S^n$ is a weighting coefficient expressing the discrete integration of the previous integral.

Using (5), the following expression is derived :

$$\mathcal{J}_{LS} = \sum_{n=1}^{N} \frac{1}{2}[{B^n_{\mathcal{M}_{l+1}}}^T \Delta P^{(0)}_{\mathcal{M}_{l+1}} - \Delta^n q^{(k_{opt})}_{\mathcal{M}_l}]^2 \delta S^n. \tag{8}$$

The solution of this problem can be easily obtained by expressing that the gradient of the functional vanishes, which yields the following linear system of normal equations :

$$\left\{ \sum_{n=1}^{N} B^n_{\mathcal{M}_{l+1}} {B^n_{\mathcal{M}_{l+1}}}^T \delta S^n \right\} \Delta P^{(0)}_{\mathcal{M}_{l+1}} = \left[ \sum_{n=1}^{N} B^n_{\mathcal{M}_{l+1}} \Delta^n q^{(k_{opt})}_{\mathcal{M}_l} \delta S^n \right]. \tag{9}$$

Finally, for a given mapping $\mathcal{M}_{l+1}$ and a current shape $S_{\mathcal{M}_l}^{(k_{opt})}$, the FFD deformation which fits $S_{\mathcal{M}_l}^{(k_{opt})}$ in a least-squares sense for the mapping $\mathcal{M}_{l+1}$ is provided by solving the previous linear system for $\Delta P_{\mathcal{M}_{l+1}}^{(0)}$.

## 2.4 Mapping parameters

The mapping producing the FFD coordinates $(s,t,u)$ from the lattice coordinates $(\xi,\eta,\zeta)$ is used to monitor the adaption process. Therefore, this mapping has to be precisely described and the adaption variables defined. To simplify the problem, the following assumptions are made :

$$s = \phi(\xi) \quad t = \psi(\eta) \quad u = \theta(\zeta). \tag{10}$$

This choice corresponds to an uncoupled relationship between the lattice coordinates and the FFD coordinates. These assumptions are reasonnable as long as the main directions of the lattice are well suited to the shape to optimize.

Then, functions $\phi$, $\psi$ and $\theta$ are supposed to be polynomials of degrees $n'_i$, $n'_j$ and $n'_k$, and are expressed in the Bernstein polynomials basis :

$$\phi(\xi) = \sum_{i=0}^{n'_i} B_i^{n'_i}(\xi)\phi_i \quad \psi(\eta) = \sum_{j=0}^{n'_j} B_j^{n'_j}(\eta)\psi_j \quad \theta(\zeta) = \sum_{k=0}^{n'_k} B_k^{n'_k}(\zeta)\theta_k. \tag{11}$$

Finally, weighting coefficients $(\phi_i)_{i=0,\ldots,n'_i}$, $(\psi_j)_{j=0,\ldots,n'_j}$ and $(\theta_k)_{k=0,\ldots,n'_k}$ are considered as adaption variables.

Initially, weighting coefficients $(\phi_i)_{i=0,\ldots,n'_i}$, $(\psi_j)_{j=0,\ldots,n'_j}$ and $(\theta_k)_{k=0,\ldots,n'_k}$ are set to linear values, in order to define the identity mapping :

$$\begin{aligned} \phi_i &= \frac{i}{n'_i} \quad i = 0, n'_i, \\ \psi_j &= \frac{j}{n'_j} \quad j = 0, n'_j, \\ \theta_k &= \frac{k}{n'_k} \quad k = 0, n'_k. \end{aligned} \tag{12}$$

Then, the adaption procedure consists in modifying the adaption variables $(\phi_i)_{i=0,\ldots,n'_i}$, $(\psi_j)_{j=0,\ldots,n'_j}$ and $(\theta_k)_{k=0,\ldots,n'_k}$ to minimize the adaption cost function.

In order to define a one-to-one mapping, functions $\phi$, $\psi$ and $\theta$ should be homeomorphisms from $[0,1]$ to $[0,1]$. Sufficient conditions to achieve this criterion can be derived from the termination properties and derivation rules of Bernstein polynomials [8]. Sufficient conditions can be expressed as :

$$\begin{aligned} &\phi_0 = \psi_0 = \theta_0 = 0, \\ &\phi_{n'_i} = \psi_{n'_j} = \theta_{n'_k} = 1, \\ &\phi_i < \phi_{i+1} \quad i = 0, n'_i - 1, \\ &\psi_j < \psi_{j+1} \quad j = 0, n'_j - 1, \\ &\theta_k < \theta_{k+1} \quad k = 0, n'_k - 1. \end{aligned} \tag{13}$$

These conditions are taken into account as linear constraints during the minimization of the adaption cost function.

The degrees $n'_i$, $n'_j$ and $n'_k$ rule the complexity of the resulting mapping. It seems useless to define a mapping whose complexity is higher than that of the deformation. Then, the degrees of the mapping $n'_i$, $n'_j$ and $n'_k$ are simply set equal to those of the deformation $n_i$, $n_j$ and $n_k$.

One can notice that the mapping defined by (10) and (11) composed with the FFD deformation (1) results in *replacing the initial deformation basis functions (Bernstein polynomials) by others deformation basis functions of higher degrees* with respect to the lattice coordinates $(\xi,\eta,\zeta)$.

## 2.5 Adaption functional

The approach used to define the adaption functional is inspired from a method developed for two-dimensional problems, based on a parameterization by Bézier curves [2, 6, 9]. In that simpler case, it was shown that an adaption process relying on the regularization of the control polygon is particularly efficient. Indeed, it was observed that the shape optimization process yields a highly irregular control polygon. This phenomenon was also demonstrated by a spectral analysis of the algebraic system in [5]. On the other hand, it is well known that if the number of control points tends to infinity, the control polygon tends to the curve, which is regular. Therefore, an irregular control polygon can be considered as an indicator of a low maturity parameterization. Then, an efficient adaption algorithm was built, that regularizes the control polygon by minimizing the total variation of the control points coordinates $\{y_i\}_{i=0,\dots,n_i}$ :

$$TV(\{y_i\}) = \sum_{i=1}^{n_i} |y_i - y_{i-1}| \approx \int_0^1 |y'(t)| dt, \tag{14}$$

where $y(t)$ is a regular function interpolating the control points $\{y_i\}_{i=0,\dots,n_i}$.

This approach is extended to FFD by introducing a measure of the variations of the deformation in each direction :

$$\int_0^1 \int_0^1 \int_0^1 \left\| \overline{\overline{\nabla}} \delta P(\xi,\eta,\zeta) \right\| d\xi \, d\eta \, d\zeta, \tag{15}$$

where $\delta P(\xi,\eta,\zeta)$ is a regular function from $[0,1]^3$ to $\Re^3$ interpolating the control points displacements and $\overline{\overline{\nabla}}$ is its gradient tensor. The formula (15) is the multidimensional extension of the integral (14).

Since $\delta P(\xi,\eta,\zeta)$ is only known at $(n_i+1)\times(n_j+1)\times(n_k+1)$ points, a discrete form of (15) is used in practice to evaluate the adaption cost function :

$$\mathcal{J}_{AD} = \frac{1}{n_i n_j n_k} \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \left\| \overline{\overline{\nabla}} \delta P_{ijk} \right\|. \tag{16}$$

$\|.\|$ is a matrix norm and $\overline{\overline{\nabla}}\delta P_{ijk}$ an estimate of $\overline{\overline{\nabla}}\delta P(\xi,\eta,\zeta)$ over the elementary volume of indices $ijk$. The use of the Frobenius norm and a finite-difference approximation of the gradient over the elementary volume of indices $ijk$ yields :

$$\begin{aligned}
\left\| \overline{\overline{\nabla}} \delta P_{ijk} \right\| = \Bigg[ & \sum_{l=1}^{3} \frac{n_i^2}{4} \Big\{ (\Delta P^l_{ijk} - \Delta P^l_{i-1jk})^2 + (\Delta P^l_{ij-1k} - \Delta P^l_{i-1j-1k})^2 \\
& + (\Delta P^l_{ijk-1} - \Delta P^l_{i-1jk-1})^2 + (\Delta P^l_{ij-1k-1} - \Delta P^l_{i-1j-1k-1})^2 \Big\} \\
& + \frac{n_j^2}{4} \Big\{ (\Delta P^l_{ijk} - \Delta P^l_{ij-1k})^2 + (\Delta P^l_{i-1jk} - \Delta P^l_{i-1j-1k})^2 \\
& + (\Delta P^l_{ijk-1} - \Delta P^l_{ij-1k-1})^2 + (\Delta P^l_{i-1jk-1} - \Delta P^l_{i-1j-1k-1})^2 \Big\} \\
& + \frac{n_k^2}{4} \Big\{ (\Delta P^l_{ijk} - \Delta P^l_{ijk-1})^2 + (\Delta P^l_{ij-1k} - \Delta P^l_{ij-1k-1})^2 \\
& + (\Delta P^l_{i-1jk} - \Delta P^l_{i-1jk-1})^2 + (\Delta P^l_{i-1j-1k} - \Delta P^l_{i-1j-1k-1})^2 \Big\} \Bigg]^{\frac{1}{2}},
\end{aligned} \tag{17}$$

where $\Delta P^l_{ijk}$ is the component $l$ of the displacement of the control point $\Delta P_{ijk}$.

Finally, the adaption procedure consists in minimizing the adaption cost function $\mathcal{J}_{AD}$ evaluated using (16) and (17), with the constraint that the current shape remains unaltered, in a least-square sense.

## 2.6 Adaption algorithm

From the developments described above, the adaption algorithm can now be defined in detail. For a current mapping defined by the adaption parameters $(\phi_i)_{i=0,\dots,n'_i}$, $(\psi_j)_{j=0,\dots,n'_j}$ and $(\theta_k)_{k=0,\dots,n'_k}$, and an current shape $S_{\mathcal{M}_l}^{(k_{opt})}$ defined by the design variables $\Delta P_{\mathcal{M}_l}^{(k_{opt})}$, the adaption procedure is described by the following algorithm:

1. Estimate the fitness $\mathcal{J}_{AD}$ of the current mapping using (15) ;
2. Begin the optimization loop ;
3. Evaluate the gradient of $\mathcal{J}_{AD}$ with respect to the adaption parameters $(\phi_i)_{i=0,\dots,n'_i}$, $(\psi_j)_{j=0,\dots,n'_j}$ and $(\theta_k)_{k=0,\dots,n'_k}$ ;
4. Update $(\phi_i)_{i=0,\dots,n'_i}$, $(\psi_j)_{j=0,\dots,n'_j}$ and $(\theta_k)_{k=0,\dots,n'_k}$ using the derivatives (gradient-based optimization) yielding a new mapping ;
5. Compute the new FFD coordinates for all nodes on $S_0^{(0)}$ using (13), (11) and (10) ;
6. Compute the new vectors $B$ defined by (3) for the tensorial product ;
7. Evaluate the least-squares approximation of $S_{\mathcal{M}_l}^{(k_{opt})}$ for the new mapping by solving the linear system (9), yielding the new control points displacements $\Delta P_{\mathcal{M}_{l+1}}^{(0)}$ ;
8. Estimate the fitness of the new mapping using (15);
9. End of the optimization loop ;
   If stopping criterion reached then stop ;
   Else goto 3.

## 2.7 Numerical implementation

The gradient of the adaption cost function $\mathcal{J}_{AD}$ with respect to the adaption variables $(\phi_i)_{i=0,\dots,n'_i}$, $(\psi_j)_{j=0,\dots,n'_j}$ and $(\theta_k)_{k=0,\dots,n'_k}$ has to be evaluated at each step of the minimization of $\mathcal{J}_{AD}$. For the sake of simplicity, this gradient is computed using a finite difference approximation.

The update of the adaption parameter $(\phi_i)_{i=0,\dots,n'_i}$, $(\psi_j)_{j=0,\dots,n'_j}$ and $(\theta_k)_{k=0,\dots,n'_k}$ to minimize $\mathcal{J}_{AD}$ is based on a steepest descent method including a line search to determine a suitable step length. Linear constraints are taken into account using projections into the feasible subspace.

The linear system (9) is solved using the LU decomposition.

# 3 A two-dimensional illustration

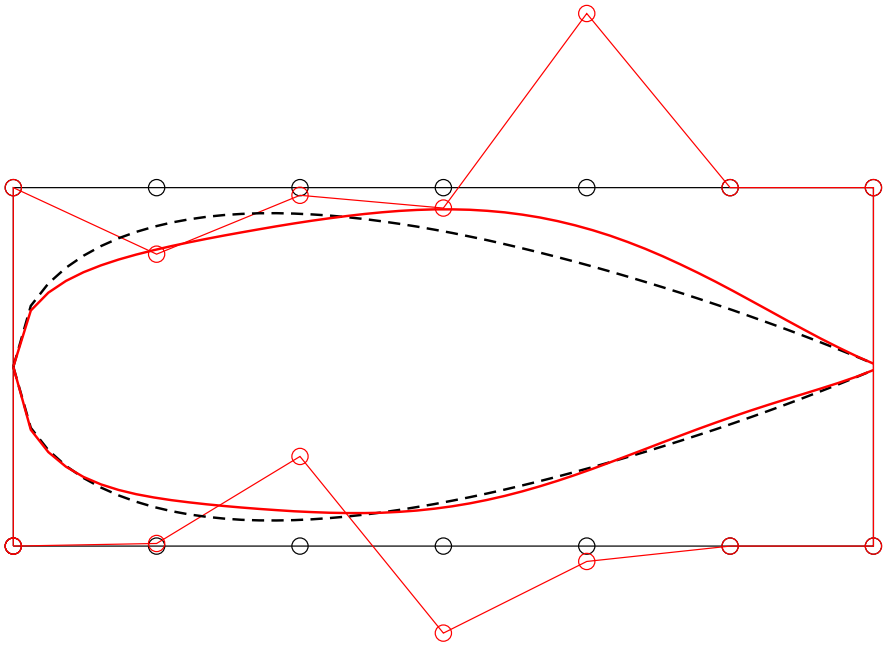
To illustrate the adaption procedure described above, a two-dimensional example shown in figure (2) is considered. Suppose that the NACA 0012 airfoil is optimized by a shape optimization procedure using a FFD parameterization ($n_i = 6$ and $n_j = 1$). This parameterization relies on the (initial and naïve) identity mapping. During the optimization, weighting coefficients for a vertical deformation are moved from zero to their optimal values. For the FFD deformation techniques, weighting coefficients can be considered as control points displacements and not control points coordinates, since the shape deformation is represented and not the shape itself. Therefore, the control points location is not strictly determined. However, it is convenient to locate the control points somewhere, in order to provide a vizualization of their displacements. Therefore, on figure (2), the control points are located at the physical points where the influence of a displacement is maximum, although these locations are somewhat arbitrary in this vizualization. In figure (2(a)), the initial FFD lattice is represented by a black line, whereas the optimized FFD lattice by a red line. These lattices correspond respectively to the initial NACA 0012 airfoil shape (black dashed curve) and the optimized airfoil shape (red curve).

Then, the adaption procedure is used to find a new mapping, which minimizes the cost function $\mathcal{J}_{AD}$, subject to the constraint that the resulting shape fits the optimized shape in a least-square sense. The history of the minimization of $\mathcal{J}_{AD}$ can be seen in figure 3. As can be seen, adaption yields a strong regularization of the FFD lattice. The resulting FFD lattice and shape are shown in figure (2(b)) (blue line and curve). As expected, the control points lattice using this new mapping is smoother than that using the identity mapping. The airfoil shape is slightly modified, mainly in the vicinity of the trailing edge. One can observe that the control points have moved horizontally since the location of the most influenced points has been modified with the mapping change. However, the regularity of the displacements of the control points is increased.

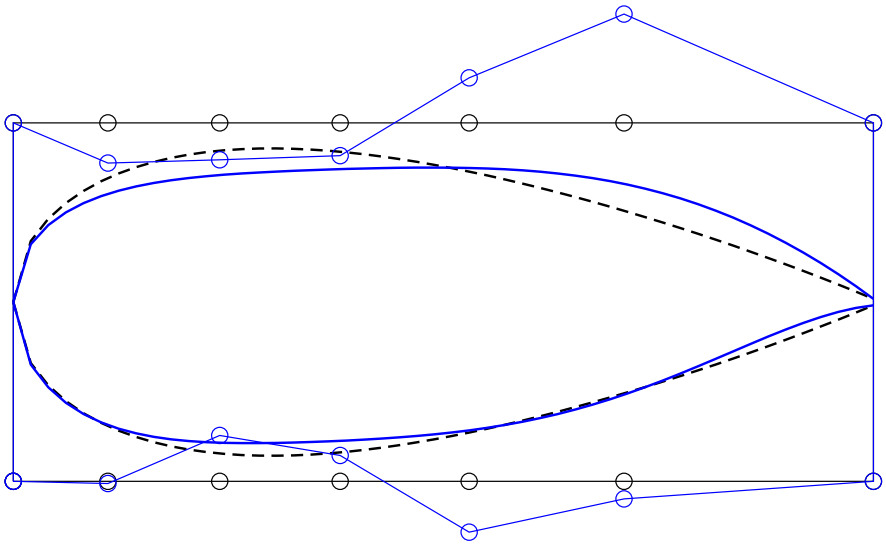
Figure (4) shows the function $\phi(\xi)$, which defines the mapping. The functions $\psi(\eta)$ and $\theta(\zeta)$ remain defined as identity since $n_j = n_k = 1$. As can be observed, adaption produces a mapping for which $s \geq \xi$. Figure (5) shows how the mapping looks like, by representing some particular isovalue contours of $s$ and $t$ in the case of the initial identity mapping (left) and the adapted mapping (right). For small $\xi$ values, $s$ isovalue contours are contracted, whereas for large $\xi$ values, they expand.

Figure (6) shows the deformation basis functions used for the FFD in terms of the $\xi$ coordinate, for the initial (dashed curves) and adapted (plain curves) mapping. The Bernstein polynomials are modified by the adaption process. Especially, their maxima are move towards smaller $\xi$ values.

The proposed approach should now be faced to a full shape optimization exercise to quantify the benefits obtained.



(a) Initial mapping (identity)



(b) Adapted mapping

Figure 2: 2D example of the adaption process. Top figure : The NACA 0012 airfoil (black dashed curve) is deformed to an optimized airfoil (red) by moving the control points of a FFD lattice from their original location (black circle markers) to their optimized location (red circle markers), for the initial mapping (identity). Bottom figure : the mapping is modified to regularize the control points lattice (blue circle markers), with the constraint that the airfoil (blue) fits in a least-square sense the optimized airfoil.

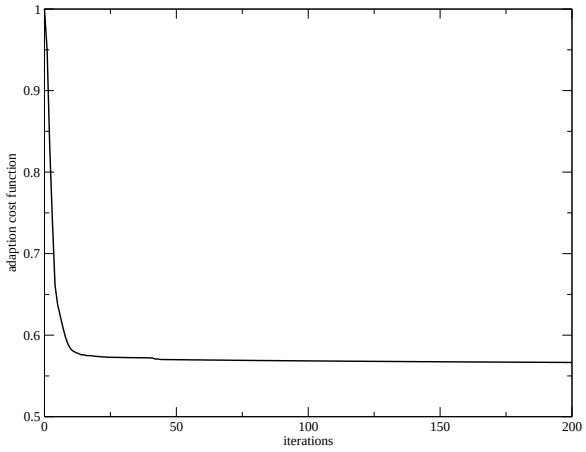



Figure 3: History of the adaption cost function $\mathcal{J}_{AD}$.

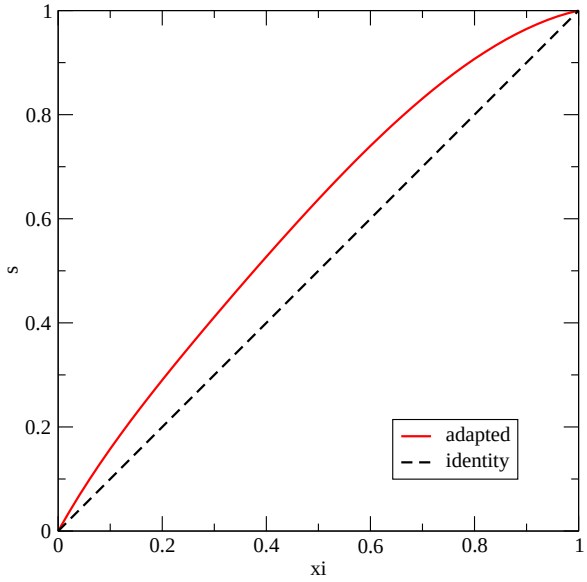
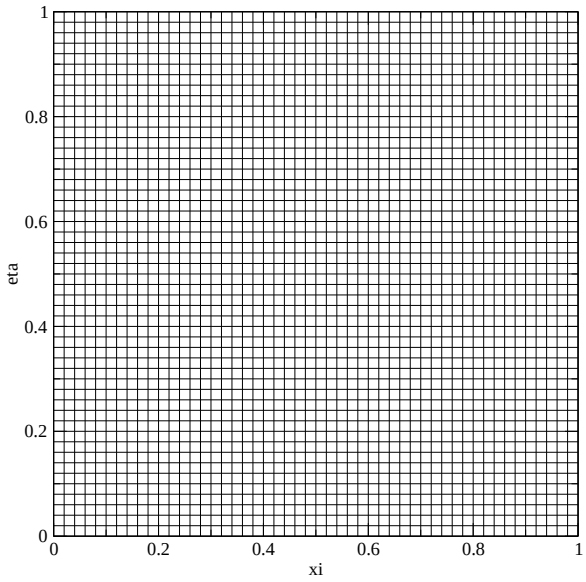
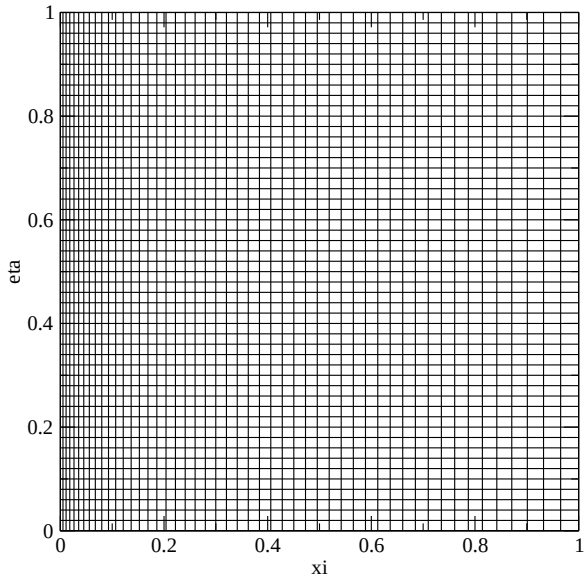



Figure 4: Mapping function $s = \phi(\xi)$.




(a) Identity mapping




(b) Adapted mapping

Figure 5: Representation of the mapping for some isovalues $(s = \phi(\xi); t = \eta))$, for the identity mapping (left) and the adapted mapping (right).

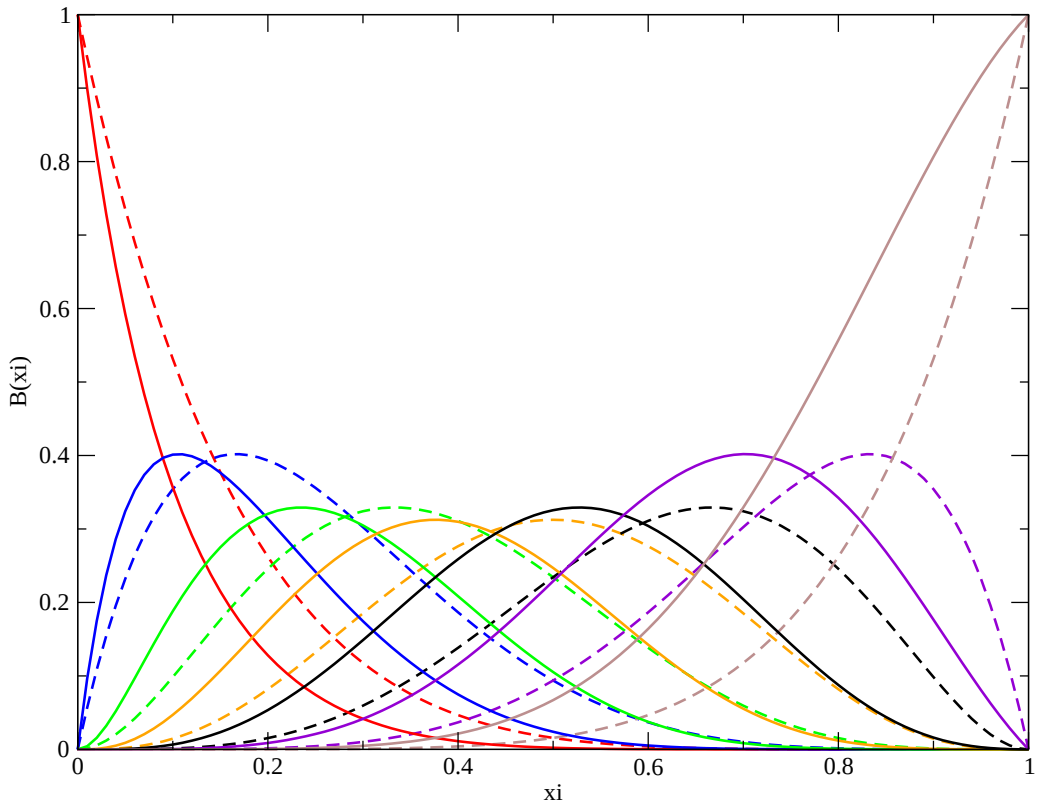



Figure 6: Deformation basis functions in terms of the $\xi$ coordinate for initial (dashed line) and adapted (plain line) mapping.

# 4 Application to 3D aerodynamic shape optimization

## 4.1 Aerodynamic shape optimization

An aerodynamic shape optimization problem consists in minimizing a cost function $\mathcal{J}_{OPT}$, which depends on design variables $x_c$ and state variables $W$. Design variables constitute a characterization of the shape to optimize. In the FFD framework, design variables are some of the $3 \times (n_i+1) \times (n_j+1) \times (n_k+1)$ weighting coefficients $\Delta P_{ijk}$ of the FFD deformation (1). State variables (i.e. flow fields) depend implicitly on the design variables through the state equations $\mathcal{E}$, which are considered as constraints. Finally, a general aerodynamic shape optimization problem can be expressed as :

$$\begin{aligned} &\text{Minimize } \mathcal{J}_{OPT}(x_c, W(x_c)), \\ &\text{Subject to } \mathcal{E}(x_c, W(x_c)) = 0, \\ &\mathcal{C}(x_c, W(x_c)) \leq 0, \end{aligned} \tag{18}$$

where $\mathcal{C}$ represents additional (physical or geometrical) constraints. A typical algorithm to solve such a problem is :

1. choose initial design variables $x_c^{(0)}$ ;
   $k \leftarrow 0$ ;
2. begin iteration $k$ of the optimization loop ;
3. update the mesh according to $x_c^{(k)}$ ;
4. solve the state equations $\mathcal{E}(x_c^{(k)}, W(x_c^{(k)})) = 0$ yielding the state variables $W(x_c^{(k)})$ ;
5. estimate the cost function $\mathcal{J}_{OPT}(x_c^{(k)}, W(x_c^{(k)}))$ ;
6. update the design variables to $x_c^{(k+1)}$ according to the optimization algorithm ;
7. finish iteration $k$ of the optimization loop ;
   if a stopping criterion is reached then STOP ;
   else $k \leftarrow k+1$ GOTO step (2).

A brief description of the flow solver, the mesh update and the optimization algorithm is provided in the next sections.

## 4.2 Flow solver

### 4.2.1 Modeling

This study is restricted to three-dimensional inviscid compressible flows governed by the Euler equations. Then, the state equations can be written in the conservative form :

$$\frac{\partial W}{\partial t} + \frac{\partial F_1(W)}{\partial x} + \frac{\partial F_2(W)}{\partial y} + \frac{\partial F_3(W)}{\partial z} = 0, \tag{19}$$

where $W$ are the conservative flow variables $(\rho, \rho u, \rho v, \rho w, E)$, with $\rho$ the density, $\overrightarrow{U} = (u, v, w)$ the velocity vector and $E$ the total energy per unit of volume. $\overrightarrow{F} = (F_1(W), F_2(W), F_3(W))$ is the vector of the convective fluxes, whose components are given by :

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix} \quad F_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix} \quad F_3(W) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}. \tag{20}$$

The pressure $p$ is obtained from the perfect gas state equation :

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \|\overrightarrow{U}\|^2), \tag{21}$$

where $\gamma = 1.4$ is the ratio of the specific heat coefficients.

#### 4.2.2 Spatial discretization

Provided that the flow domain $\Omega$ is discretized by a tetrahedrization $\mathcal{T}_h$, a discretization of equation (19) at the mesh node $s_i$ is obtained by integrating (19) over the volume $C_i$, that is built around the node $s_i$ by joining barycenters of the tetrahedra and triangles containing $s_i$ and midpoints of the edges adjacent to $s_i$ :

$$Vol_i \frac{\partial W_i}{\partial t} + \sum_{j \in N(i)} \Phi(W_i, W_j, \overrightarrow{\sigma}_{ij}) = 0, \tag{22}$$

where $W_i$ represents the cell averaged state and $Vol_i$ the volume of the cell $C_i$. $N(i)$ is the set of the neighboring nodes. $\Phi(W_i, W_j, \overrightarrow{\sigma}_{ij})$ is an approximation of the integral of the fluxes (20) over the boundary $\partial C_{ij}$ between $C_i$ and $C_j$, which depends on $W_i$, $W_j$ and $\overrightarrow{\sigma}_{ij}$ the integral of a unit normal vector over $\partial C_{ij}$. These numerical fluxes are evaluated using upwinding, according to the approximate Riemann solver of Roe :

$$\Phi(W_i, W_j, \overrightarrow{\sigma}_{ij}) = \frac{\overrightarrow{F}(W_i) + \overrightarrow{F}(W_j)}{2} \cdot \overrightarrow{\sigma}_{ij} - |A_R(W_i, W_j, \overrightarrow{\sigma}_{ij})| \frac{W_j - W_i}{2}. \tag{23}$$

$A_R$ is the jacobian matrix of the fluxes for the Roe average state and verifies:

$$A_R(W_i, W_j, \overrightarrow{\sigma}_{ij})(W_j - W_i) = (\overrightarrow{F}(W_j) - \overrightarrow{F}(W_i)) \cdot \overrightarrow{\sigma}_{ij}. \tag{24}$$

A high order scheme is obtained by interpolating linearly the physical variables from $s_i$ to the midpoint of $[s_i s_j]$, before equation (22) is employed to evaluate the fluxes. Nodal gradients are obtained from a weighting average of the P1 Galerkin gradients computed on each tetrahedron containing $s_i$. In order to avoid spurious oscillations of the solution in the vicinity of the shock, a slope limitation procedure using the Van-Albada limiter is introduced. The resulting discretization scheme exhibits a third order accuracy in the regions where the solution is regular.

#### 4.2.3 Time integration

A first order implicit backward scheme is employed for the time integration of (22), which yields :

$$\frac{Vol_i}{\Delta t} \delta W_i + \sum_{j \in N(i)} \Phi(W_i^{n+1}, W_j^{n+1}, \overrightarrow{\sigma}_{ij}) = 0, \tag{25}$$

with $\delta W_i = W_i^{n+1} - W_i^n$. Then, the linearization of the numerical fluxes provides the following integration scheme :

$$\left( \frac{Vol_i}{\Delta t} + J_i^n \right) \delta W_i = - \sum_{j \in N(i)} \Phi(W_i^n, W_j^n, \overrightarrow{\sigma}_{ij}). \tag{26}$$

Here, $J_i^n$ is the jacobian matrix of the first order numerical fluxes, whereas the right hand side of (26) is evaluated using high order approximations. The resulting integration scheme provides a high order solution of the problem. More details can be found in [4].

### 4.3 Mesh update

For each step of the shape optimization procedure, the flow domain has to be modified since its boundary corresponding to the shape to optimize is deformed. Then, a new grid should be generated in accordance with the boundary displacement.

Suppose that the FFD technique is used to parameterize the shape deformation. Since FFD is a technique that deforms the space, *FFD can be used to deform the mesh and the shape simultaneously.* However, one should proceed carefully, in order to ensure that a smooth deformation is imposed, that does not generate overlapping effects in the mesh. The smoothness of the deformation is ensured inside the lattice, since deformation is ruled by Bernstein polynomials. A special emphasis should be given to the deformation smoothness at the boundary of the lattice, since no deformation occurs outside the lattice.

A feasible approach consists in introducing a blending function $\omega$, whose role is to smooth the deformation at the boundary of the FFD lattice. The blended FFD deformation is defined as :

$$\Delta q = \omega(\xi_q)\omega(\eta_q)\omega(\zeta_q) \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} B_i^{n_i}(s_q) B_j^{n_j}(t_q) B_k^{n_k}(u_q) \Delta P_{ijk}. \tag{27}$$

A so-called blending lattice is defined inside the FFD lattice. In this blending lattice, the blending function $\omega$ is equal to one. Then, FFD deformation proceeds as usually and yields a smooth deformation of the shape and the mesh simultaneously. Outside the blending lattice, the blending function decreases regularly to zero at the boundary of the FFD lattice. Outside the FFD lattice, its value is zero. An appropriate choice of the blending function can ensure the $C^\infty$-continuity of the deformation over the whole computational domain. For instance, consider the blending function:

$$\omega(\tau) = \begin{cases} 0 & \text{if } \tau \leq \tau_{min}^{ffd}, \\ \frac{1}{2}\left[1 + \tanh\left\{\alpha\left(\tau_{min}^{ffd} - \tau_{min}^{bl}\right)\left(\frac{1}{\tau - \tau_{min}^{bl}} + \frac{1}{\tau - \tau_{min}^{ffd}}\right)\right\}\right] & \text{if } \tau \in ]\tau_{min}^{ffd}, \tau_{min}^{bl}[, \\ 1 & \text{if } \tau \in [\tau_{min}^{bl}, \tau_{max}^{bl}], \\ \frac{1}{2}\left[1 + \tanh\left\{\alpha\left(\tau_{max}^{ffd} - \tau_{max}^{bl}\right)\left(\frac{1}{\tau - \tau_{max}^{bl}} + \frac{1}{\tau - \tau_{max}^{ffd}}\right)\right\}\right] & \text{if } \tau \in ]\tau_{max}^{bl}, \tau_{max}^{ffd}[, \\ 0 & \text{if } \tau \geq \tau_{max}^{ffd}, \end{cases} \tag{28}$$

where $\tau_{min}^{ffd}$, $\tau_{max}^{ffd}$, $\tau_{min}^{bl}$ and $\tau_{max}^{bl}$ are the coordinates of the boundaries of the FFD and blending lattices along the direction $\tau$. Then, the function $\omega$ is $C^\infty$ over $\Re$ and (28) yields the $C^\infty$-continuity of the deformation over the whole computational domain since discontinuities can only be found at the boundary of the FFD lattice, where all derivatives of $\omega$ vanish. The blending function (28) is depicted in figure (7). The slope of the proposed

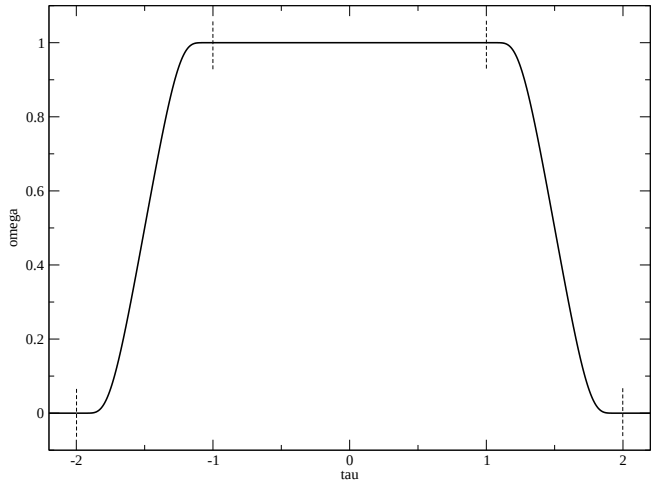



Figure 7: Example of blending function for $\tau_{min}^{ffd} = -2$, $\tau_{max}^{ffd} = 2$, $\tau_{min}^{bl} = -1$ and $\tau_{max}^{bl} = 1$

function (28) can be controlled by the numerical parameter $\alpha$. The influence of $\alpha$ on the blending function is shown in figure (8). For large values, the blending function tends to the Heaviside function, whereas for low values the slope remains bounded. In practice, a low value is preferred, yielding a more progressive blending effect.

Finally, the efficiency of such a grid deformation technique is demonstrated for a two-dimensional unstructured mesh. The mesh around the NACA 0012 airfoil is deformed according to the technique described above. The FFD deformation is imposed by moving some control points vertically, in order the shape to fit an optimized airfoil. Figure (9) shows the inital (blue) and deformed (red) meshes, as well as the FFD (cian) and blending (green) lattices. As can be seen, the mesh is only deformed inside the FFD lattice and this deformation tends regularly to zero outside the blending lattice. The grids in the vicinity of the airfoil can be seen in figures (10(a)) and (10(b)).

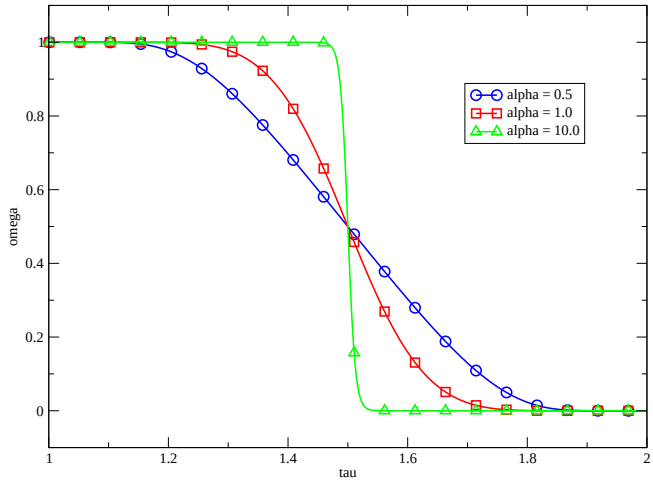



Figure 8: Influence of the parameter $\alpha$

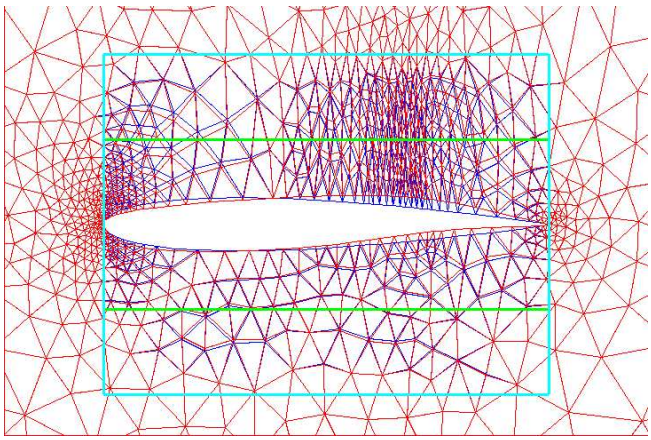
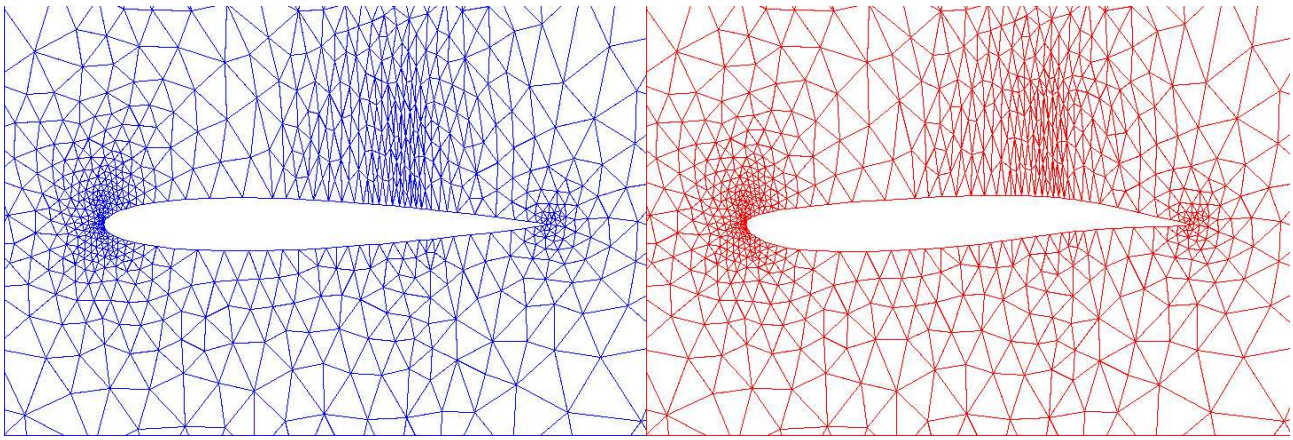


Figure 9: Example of FFD (cian) and blending (green) lattices.



(a) Initial grid (b) Deformed grid

Figure 10: Example of grid deformation.

## 4.4 Optimization algorithm

The optimization method used to lead the search is the Multi-directional Search Algorithm (MSA) developped by Torczon [3, 13]. This algorithm is inspired from the Nelder-Mead simplex method. Torczon proposed modifications to correct some of its drawbacks. Particularly, a proof of convergence is given under classical assumptions concerning the regularity of the cost function [3, 13]. Moreover, the algorithm is designed to take advantage of a parallel architecture.

The algorithm consists in moving a simplex of $n+1$ vertices $(v_i)_{i=0,...,n}$ in $\Re^n$ (a triangle in $\Re^2$, a tetrahedron in $\Re^3$, etc), for a problem of $n$ design parameters. Each vertex represents a different design. The simplex is first initialized, by perturbing for instance the initial design parameters $x_c^{(0)}$ in each of the $n$ directions $e_i$ successively:

$$v_0^{(0)} = x_c^{(0)} \quad v_i^{(0)} = x_c^{(0)} + \delta e_i \quad i = 1, ..., n. \tag{29}$$

The amplitude of the initial perturbation $\delta$ is problem dependent. After the cost function is evaluated at all vertices, displacements are performed in order to decrease the cost function. At each step $k$, assume that $v_0^{(k)}$ is the best vertex of the current simplex. Then, the whole simplex is reflected with respect to $v_0^{(k)}$ :

$$v_i^r = (1+\alpha)v_0^{(k)} - \alpha v_i^{(k)} \quad i = 1, ..., n, \tag{30}$$

with $\alpha$ set to unity. In case of success, i.e. if the cost function evaluated at any of the vertices $v_1^r$, ... , $v_n^r$ is lower than that evaluated at $v_0^{(k)}$, the simplex is expanded from $v_0^r$ :

$$v_i^e = \gamma v_i^r + (1-\gamma)v_0^r \quad i = 1, ..., n, \tag{31}$$

with $\gamma$ set to two. On the contrary, in case of failure, i.e. if the cost function evaluated at all vertices $v_1^r$, ... , $v_n^r$ is higher than that evaluated at $v_0^{(k)}$, the simplex is contracted from $v_0^r$ :

$$v_i^c = \beta v_i^r + (1-\beta)v_0^r \quad i = 1, ..., n, \tag{32}$$

with $\beta$ set to minus half. the three basic movements are depicted on figure (11). Then, the algorithm goes on

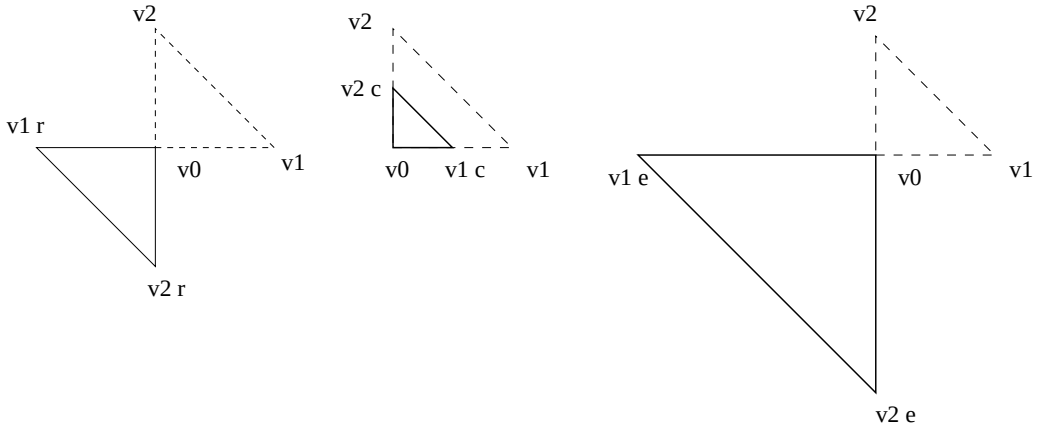



Figure 11: Example of reflection (left), contraction (center) and expansion (right).

with the new simplex for the step $k+1$. The loop is repeated until a prescribed stopping criterion is reached. In the present work, the stopping criterion is based on the size of the edge from the best vertex $v_0^{(k)}$ to other vertices.

The above algorithm is well suited to parallel computing, since twice $n$ independent and simultaneous evaluations of the cost function have to be performed at each iteration. Then, the algorithm can be implemented easily on a parallel architecture.

# 5 Results

## 5.1 Test-case description

The test-case considered here corresponds to the optimization of the shape of the wing of a business aircraft (courtesy of Piaggio Aero Industries) for a transonic regime. The test-case is described in depth in [1]. The free-stream Mach number is $M_\infty = 0.83$ and the incidence $\alpha = 2°$. Initially, the wing section is supposed to correspond to the NACA 0012 airfoil. An unstructured mesh, composed of 31124 nodes and 173 445 elements, is generated around the wing, including a refined area in the vicinity of the shock (figure (12)).

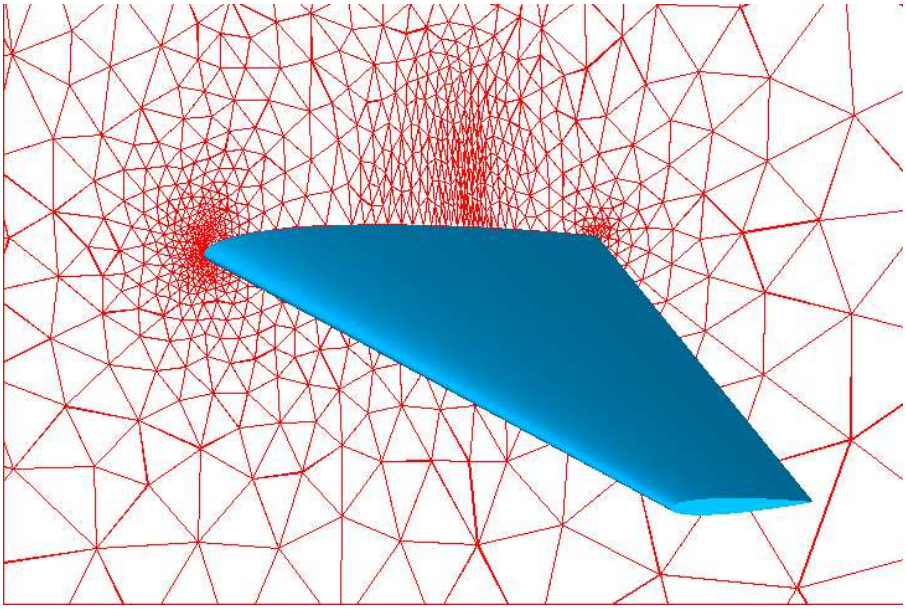


Figure 12: Initial wing shape (blue) and mesh (red) in the symetry plane.

The goal of the optimization is to reduce the drag coefficient $C_D$ subject to the constraint that the lift coefficient $C_L$ should not decrease more than 0.1%. The constraint is taken into account using a penalization approach. Then, the resulting cost function is :

$$\mathcal{J}_{OPT} = \frac{C_D}{C_{D0}} + 10^4 \max(0, 0.999 - \frac{C_L}{C_{L0}}). \tag{33}$$

$C_{D0}$ and $C_{L0}$ are respectively the drag and lift coefficients corresponding to the initial shape (NACA 0012 section).

The FFD lattice is built around the wing with $\xi$, $\eta$ and $\zeta$ in the chordwise, spanwise and thickness directions respectively. The lattice is chosen in order to fit the planform of the wing (see figure 1). Then, the leading and trailing edges are kept fixed during the optimization by freezing the control points that correspond to $i = 0$ and $i = n_i$. Moreover, control points are only moved vertically. Results are presented for three parameterizations. The coarsest one corresponds to $n_i = 3$, $n_j = 1$ and $n_k = 1$. Therefore, $(4-2) \times 2 \times 2 = 8$ degrees of freedom are taken into account in the optimization. The medium parameterization corresponds to $n_i = 6$, $n_j = 1$ and $n_k = 1$ and counts $(7-2) \times 2 \times 2 = 20$ degrees of freedom. Finally, the finest parameterization corresponds to $n_i = 9$, $n_j = 1$ and $n_k = 1$ and counts $(10-2) \times 2 \times 2 = 32$ degrees of freedom.

To assess the adaption procedure, the optimization problem is first solved until convergence using the initial and naïve parameterization, before adaption occurs. Therefore, the optimal shape in the framework of the initial parameterization drives the adaption procedure. The question of the most efficient time to carry out adaption is left for future work.

## 5.2 Results for adaption at convergence with re-initialization

The first computational experiments involve the coarse parameterization of the deformation. The shape optimization problem is first solved using the identity mapping, providing an approximation of the optimal shape. It

is then employed to adapt the mapping, before a new optimization problem is solved using the adapted parameterization and re-initializing the problem from the initial shape (NACA 0012 section). The histories of the cost functions during both optimization exercises are depicted in figure (13). As can be seen, the parameterization adaption yields a slower convergence towards a low performance design.

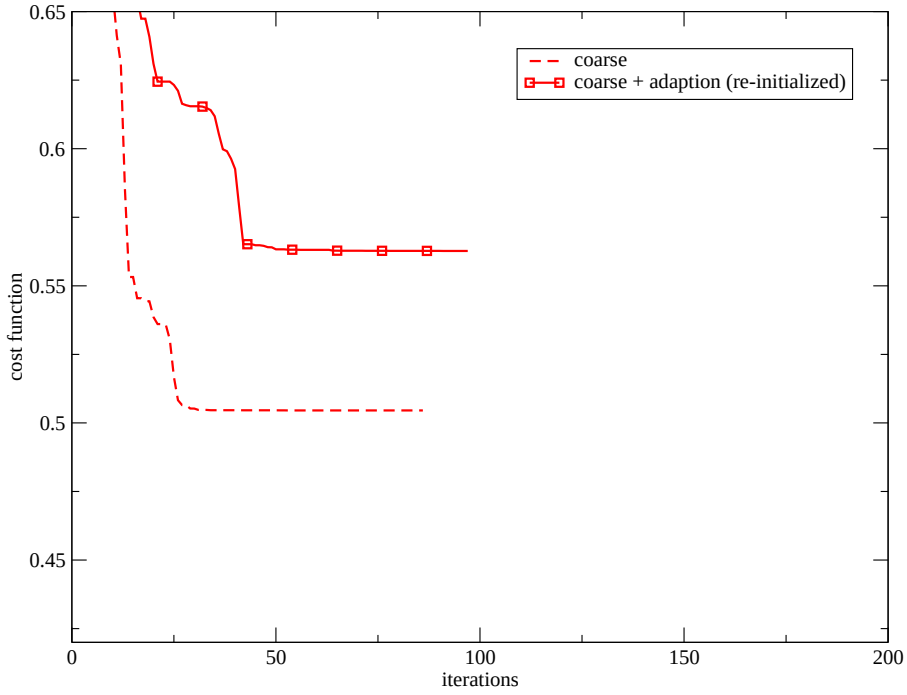



Figure 13: History of the cost function : coarse parameterization using the MSA method, with and without adaption, with re-initialization.

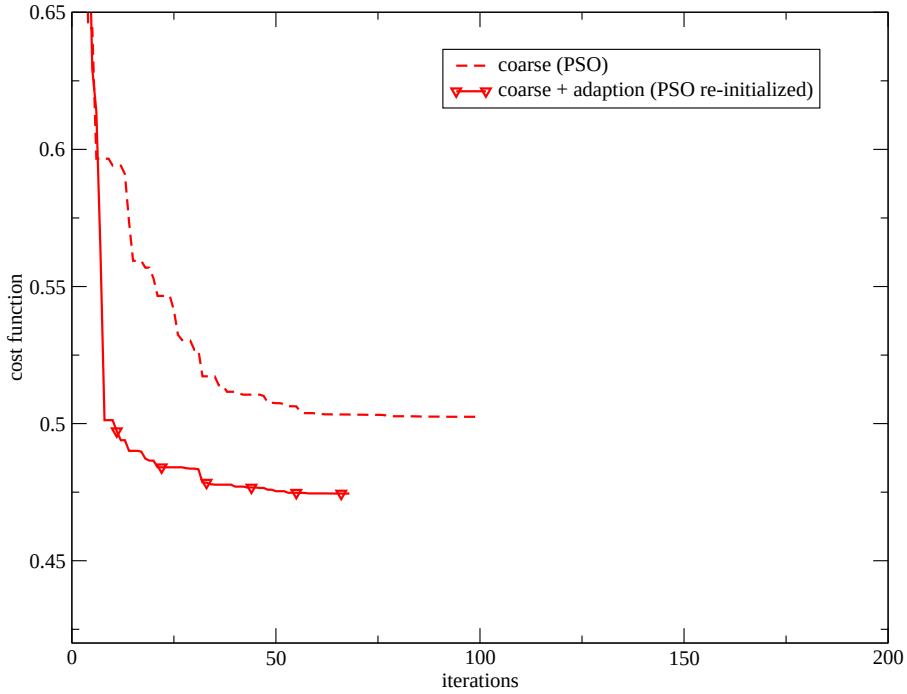



Figure 14: History of the cost function : coarse parameterization using the PSO algorithm, with and without adaption, with re-initialization.

This poor result is due to either a bad choice of the adaption strategy or a local optimization. To assess these assumptions, the same exercise is carried out using a semi-stochastic optimizer (Particle Swarm Optimization [14]), whose results do not depend on a starting point. As using the MSA method, a first optimization exercise is performed until convergence before adaption and a second optimization exercise are carried out. The histories of the cost functions during both optimization exercises are depicted in figure (14). As can be seen, the best cost function value obtained with the initial parameterization is slightly lower than that obtained with the MSA method. This result is not surprising, considering the global search capability of the stochastic optimizer. Moreover, the results are significantly improved using the adapted parameterization. The convergence is faster and the best cost function value obtained is far lower than that reached with the initial parameterization.

These two first tests show that adaption can improve the conditioning of the optimization problem and the accuracy of the results obtained, but it yields large modifications of the functional which can generate local optima.

## 5.3 Results for adaption at convergence with continuation

To benefit from the adaption procedure without using a stochastic optimizer, the optimization after adaption will now proceed without re-initializing the optimization problem from the starting design. Then, the starting point for the optimizer using the adapted parameterization will be the least-squares approximation of the best design found using the previous parameterization.

### 5.3.1 Coarse parameterization

First, this new strategy is tested using the coarse parameterization described above. Two adaption + optimization steps are successively performed after a first optimization with the initial parameterization. The history of the cost function is depicted in figure (15).

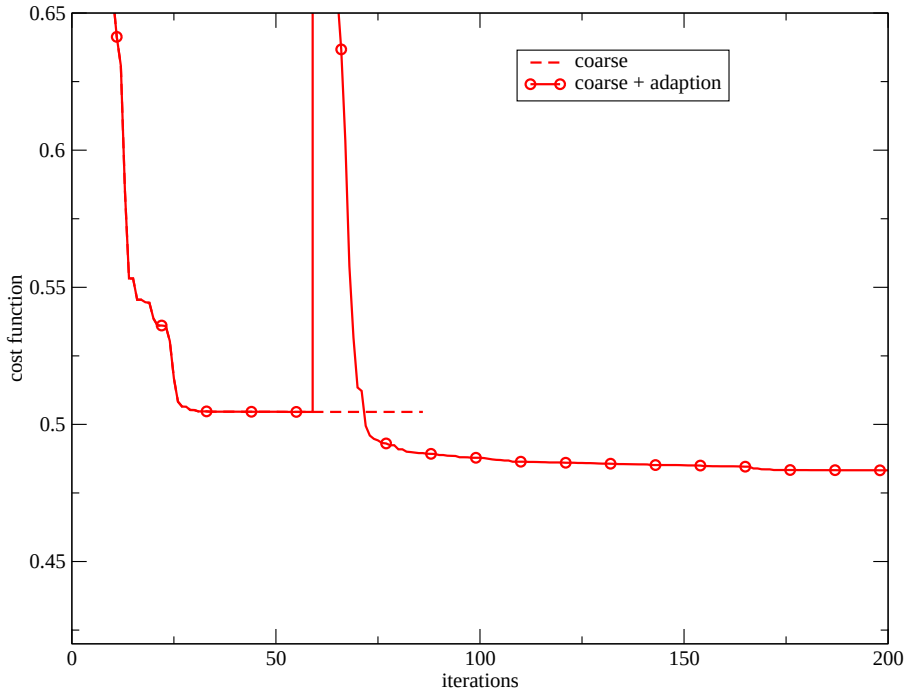



Figure 15: History of the cost function : coarse parameterization using the MSA method, with and without adaption.

As adaption occurs, a strong jump of the cost function value is observed. This is due to the fact that the starting point for the optimizer using the adapted parameterization is the least-squares approximation of the best shape found using the initial parameterization. Indeed, the optimal design found is usually close to the boundary of the feasible space. However, its least-squares approximation using the adapted mapping is outside the feasible space, which results in a penalization of the cost function. Figure (16) shows a comparison of the optimal shape found with the initial parameterization and its least-squares approximation using the adapted

parameterization. As can be seen, these shapes are close to each other, except in the vicinity of the tip section, where some discrepancies are observed. However, this difference is large enough to yield a violation of the lift constraint. Nevertheless, a feasible design is obtained after some optimization steps and the procedure converges finally to a far better design. One can notice that the second adaption step does not yield a contraint violation but again improves slightly the results.

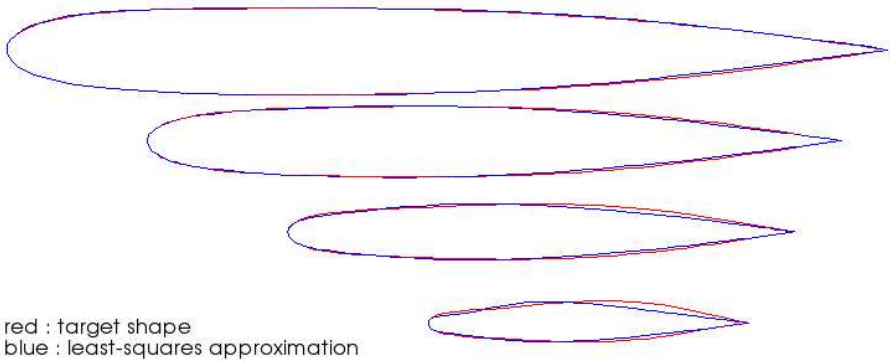



Figure 16: Comparison of the target shape $S_l^{(k_{opt})}$ (red) and its least-squares approximation with the new mapping $S_{l+1}^{(0)}$ (blue).

Figure (17) presents a comparison of the results obtained using the coarse parameterization (with and without adaption) and using the medium parameterization (without adaption) in terms of number of evaluations. Obviously, it is advantageous to choose a coarser but adapted parameterization.

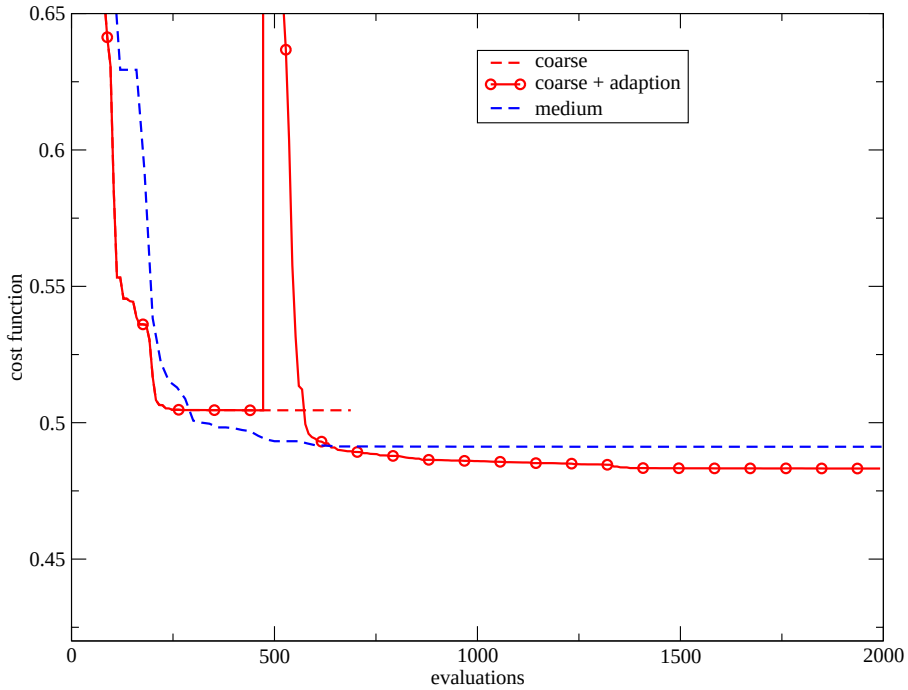



Figure 17: History of the cost function w.r.t. the number of evaluations : coarse parameterization using the MSA method, with and without adaption.

### 5.3.2 Medium parameterization

The results obtained using the medium parameterization (20 design variables) are similar to those obtained using the coarse one. The history of the cost function (figure (18)) shows that the two successive adaption procedures yield significant cost function decresases. As previously, the least-squares approximation of the

optimal shape found using the initial parameterization violates the lift constraint. However, the discrepancy between the optimal shape and its approximation is lower than that using the coarsest parameterization. Then, only two iterations are required to move back into the feasible space.

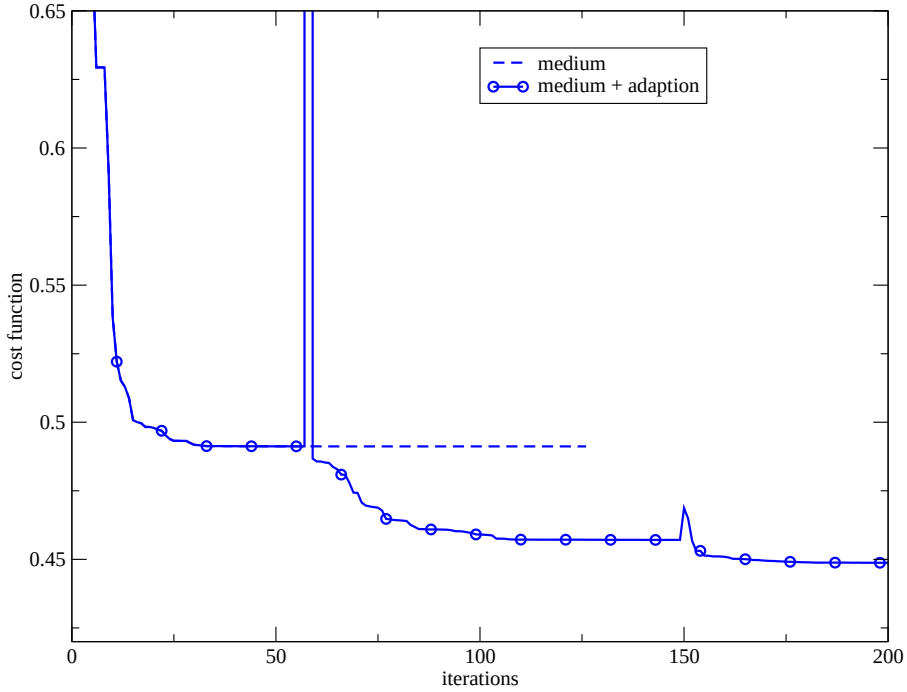



Figure 18: History of the cost function : medium parameterization using the MSA method, with and without adaption.

Figure (19) presents a comparison of the results obtained using the medium parameterization (with and without adaption) and using the fine parameterization (without adaption) in terms of number of evaluations. Contrary to the previous case, the advantage of choosing the medium but adapted parameterization is not so clear. Indeed, a better fitness is reached using the fine parameterization, although the convergence is slower.

### 5.3.3 Fine parameterization

Using the fine parameterization (32 design variables), the benefit of using adaption is moderate (figure (20)). However, the cost function decrease thanks to adaption is not negligible, providing that the parameterization is fine.

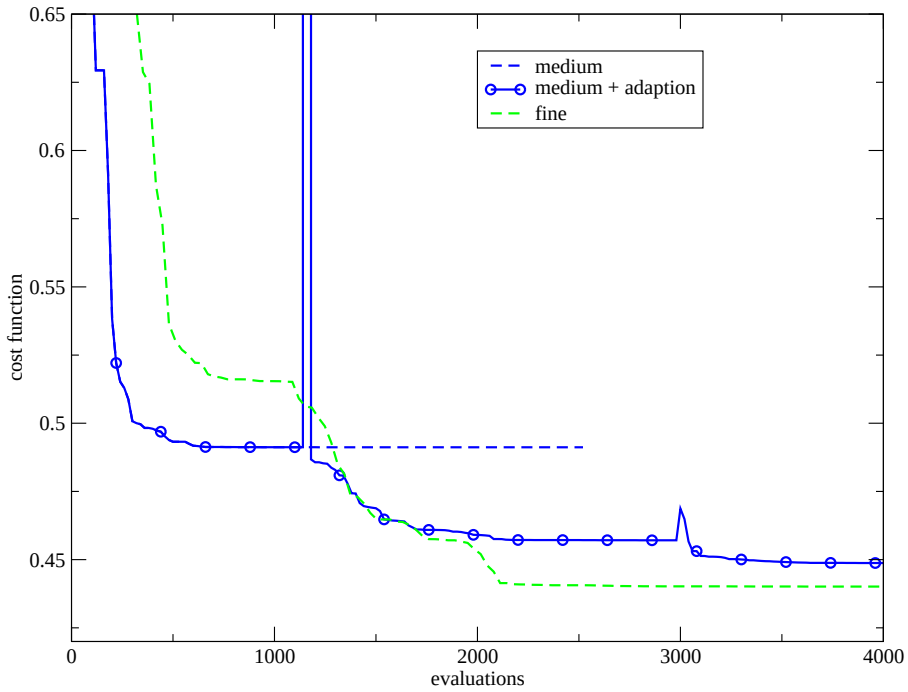



Figure 19: History of the cost function w.r.t. the number of evaluations : medium parameterization using the MSA method, with and without adaption.

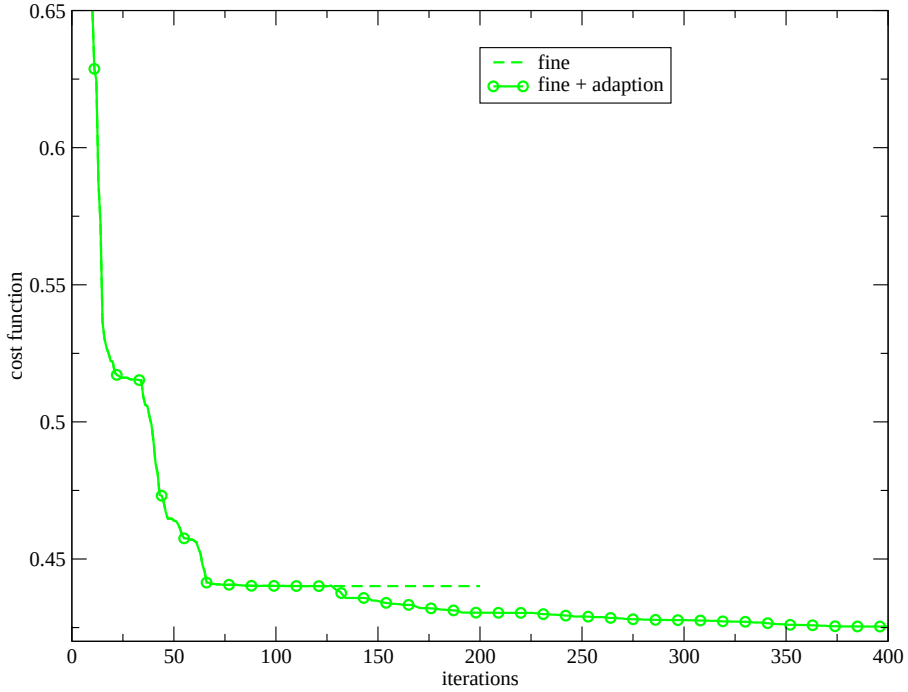



Figure 20: History of the cost function : fine parameterization using the MSA method, with and without adaption.

## 5.4 Results for a progressive adaption

The previous results assess the efficiency of the proposed adaption procedure, since it yields a significant increase of the fitness of the optimal design found. However, the advantage of choosing such a strategy is not so clear, as seen in figures representing the cost function decrease with respect to the number of evaluations (for instance figure (19)). However, this may be due to the fact that adaption is performed when the optimisation problem is almost converged, which is time consuming. The possibility to perform adaption before the full convergence of the optimization is studied in the next sections.

### 5.4.1 Coarse parameterization

The coarse parameterization is used to study to what extent it is possible to carry out adaption at a prescribed frequency. In that case, adaption is used as an update of the parameterization that is performed regularly. Actually, a simultaneous convergence of the shape optimization procedure and the parameterization procedure is aimed at.

In the previous case, two adaptions are performed as the optimization is almost converged. Three cases are now studied, for which adaption is respectively performed every 40 iterations, 20 iterations and 10 iterations. The convergence histories for these three cases are depicted in figures (21) to (22). As can be seen, results using adaption every 40 iterations are very close to those using adaption at convergence (figure (21)). Nevertheless, it proves that adaption can be carried out before the full convergence is reached. When adaption is performed every 20 iterations (figure (22)) or every 10 iterations (figure (23)), results differ significantly. Adaption permits a faster decrease of the cost function to a shape of better fitness. One can conjecture that an early and progressive adaption prevents from converging to a local optimum.

Figure (26) presents a comparison of the results obtained using the coarse parameterization (with and without progressive adaption) and using the medium parameterization (without adaption) in terms of number of evaluations. The benefit of using such an adaption strategy is obvious.

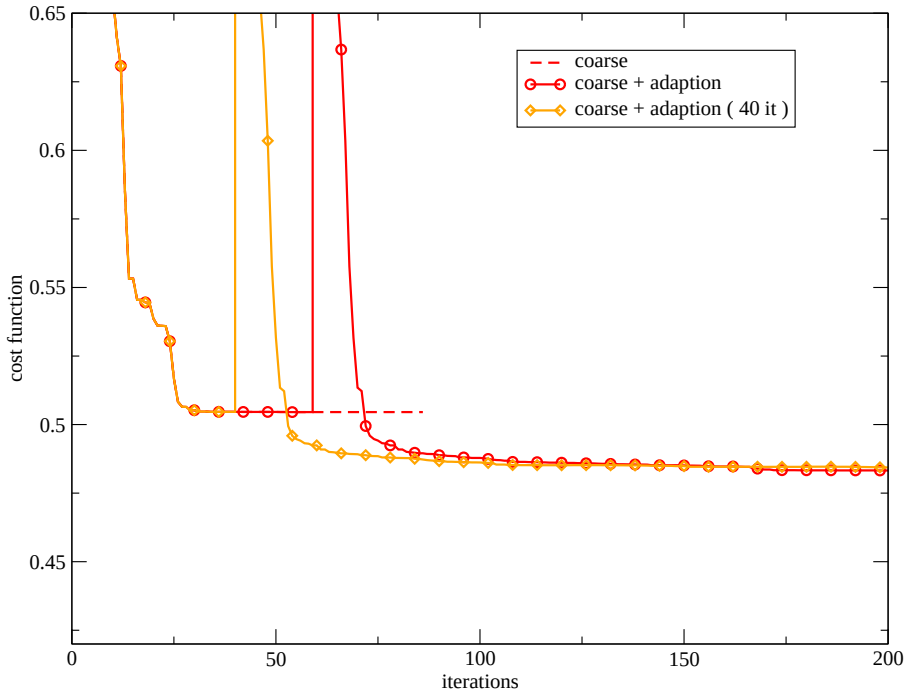



Figure 21: History of the cost function : coarse parameterization using the MSA method, with and without progressive adaption (every 40 iterations).

### 5.4.2 Medium parameterization

The same strategy is tested for the medium parameterization, using adaption every ten iterations (figure (25)). The satisfactory results obtained with the coarse parameterization are confirmed with the medium one. More-

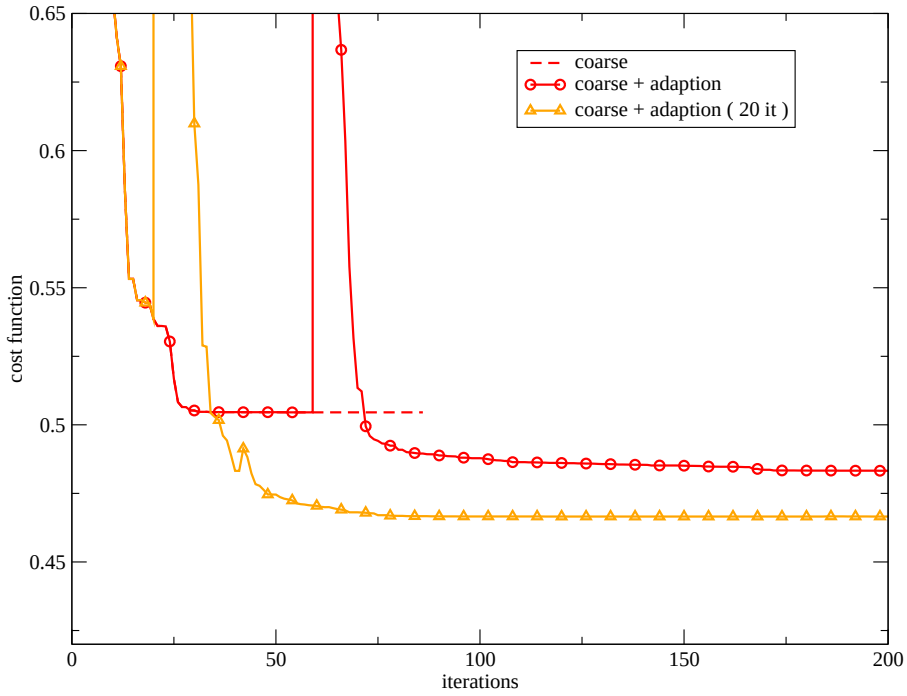



Figure 22: History of the cost function : coarse parameterization using the MSA method, with and without progressive adaption (every 20 iterations).

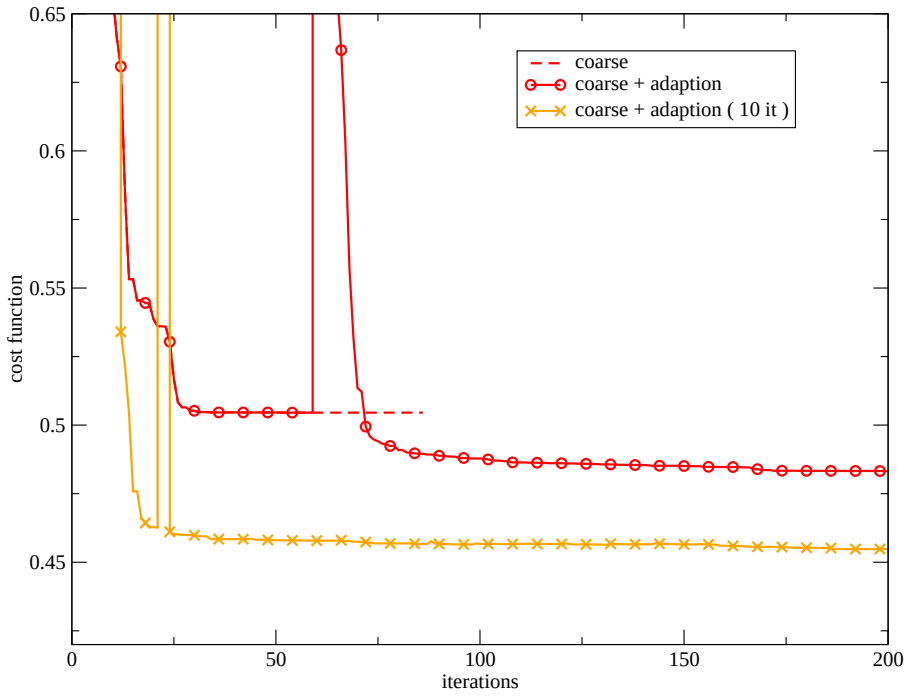



Figure 23: History of the cost function : coarse parameterization using the MSA method, with and without progressive adaption (every 10 iterations).

over, when these results are compared to those using a fine parameterization without adaption (figure (26)), one concludes that the adapted medium parameterization permits to reach a shape of fitness similar to that obtained with the fine parameterization, for half of the computational cost.

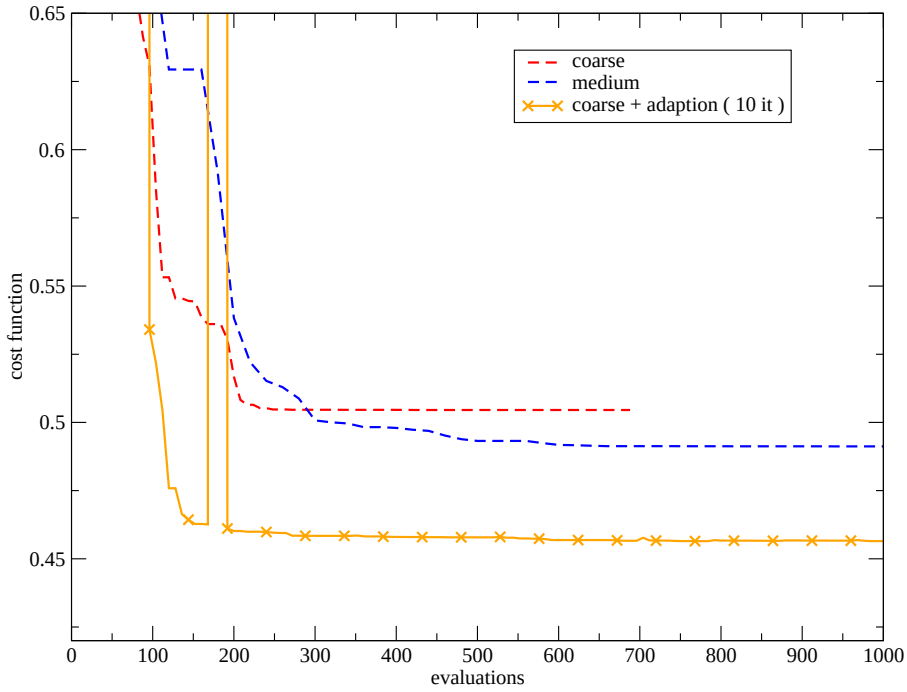



Figure 24: History of the cost function w.r.t. the number of evaluations : coarse parameterization using the MSA method, with and without progressive adaption (every 10 iterations).

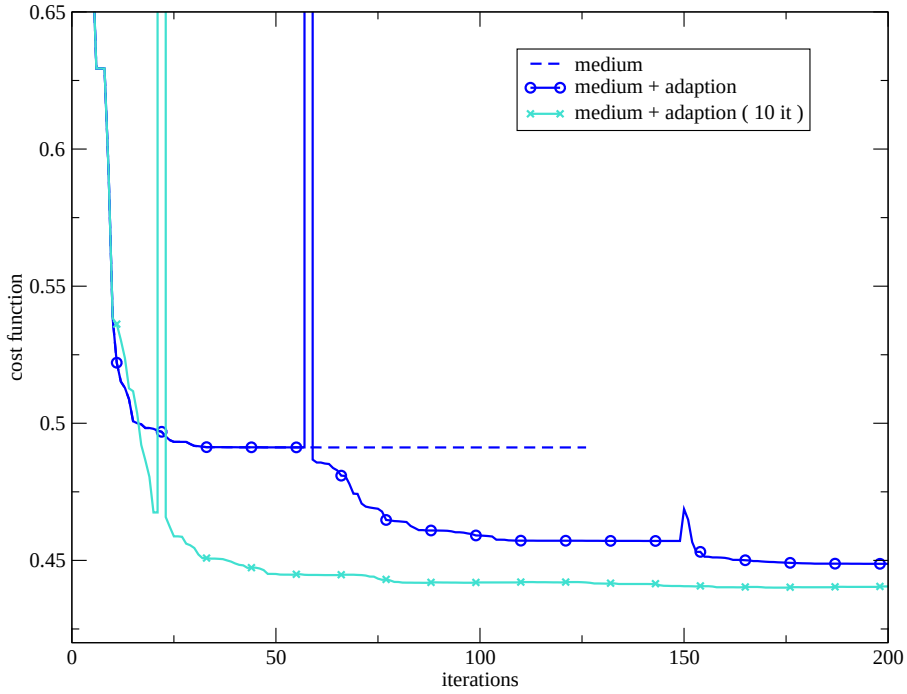



Figure 25: History of the cost function : medium parameterization using the MSA method, with and without progressive adaption (every 10 iterations).

### 5.4.3 Fine parameterization

Finally, this adaption strategy is tested for the fine parameterization. For an adaption update every 10 iterations (figure (27)), poor results are found. One can suppose that adaption is performed to early in order the shape

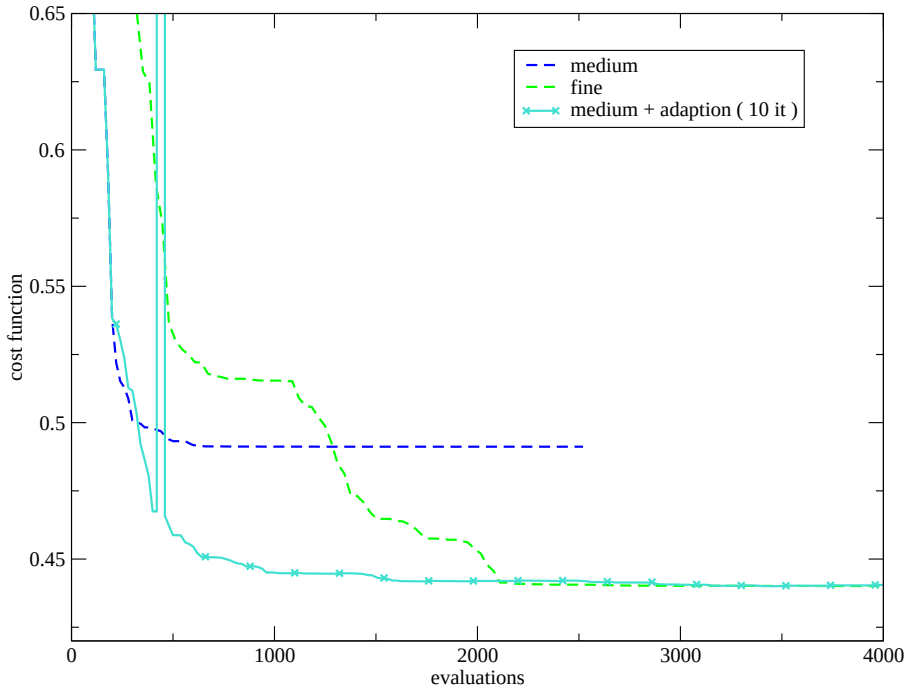



Figure 26: History of the cost function w.r.t. the number of evaluations : medium parameterization using the MSA method, with and without progressive adaption (every 10 iterations).

to provide suitable information for parameterization update. However, when adaption is carried out every 20 iterations (figure (28)), satisfactory results are obtained.

The best fitness obtained for all the computations are summarized in table (1).

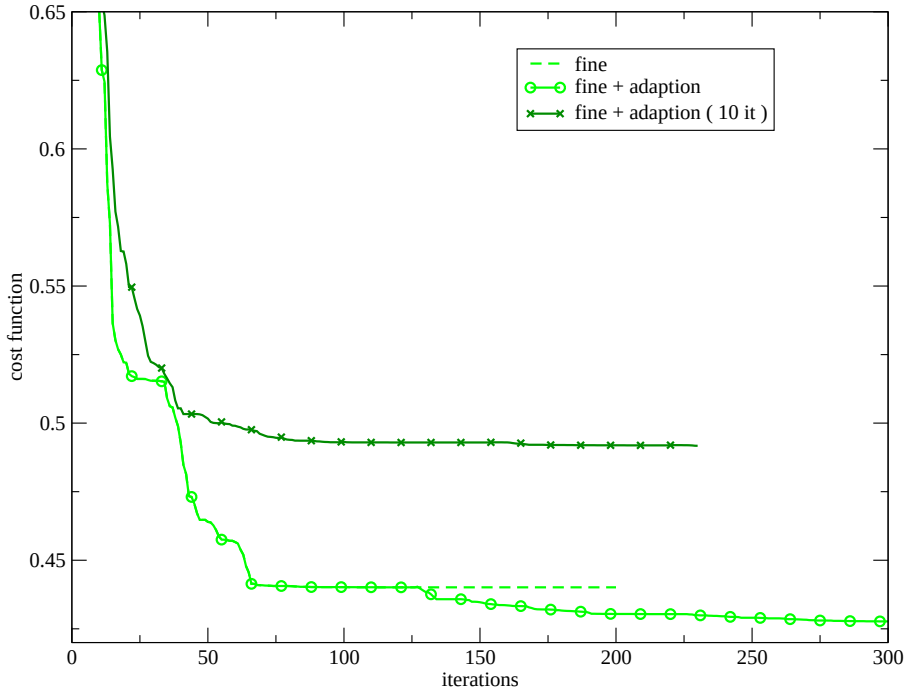



Figure 27: History of the cost function : fine parameterization using the MSA method, with and without progressive adaption (every 10 iterations).

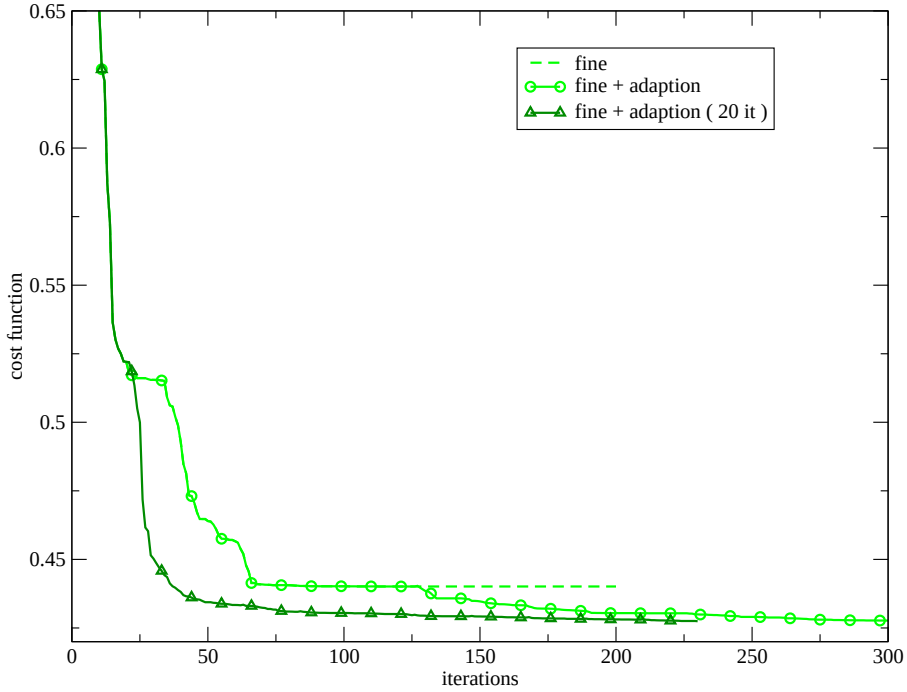



Figure 28: History of the cost function : fine parameterization using the MSA method, with and without progressive adaption (every 20 iterations).

| parameterization | adaption type | cost function |
|---|---|---|
| coarse | no adaption | 0.504 |
| coarse | at convergence | 0.483 |
| coarse | every 40 iterations | 0.484 |
| coarse | every 20 iterations | 0.466 |
| coarse | every 10 iterations | 0.455 |
| medium | no adaption | 0.491 |
| medium | at convergence | 0.449 |
| medium | every 10 iterations | 0.440 |
| fine | no adaption | 0.440 |
| fine | at convergence | 0.425 |
| fine | every 20 iterations | 0.491 |
| fine | every 10 iterations | 0.427 |

Table 1: Cost function values reached for the different computations.

## 5.5 Specific results for a medium parameterization

In the previous sections, it was shown that adaption can improve the efficiency of the shape optimization procedure. Now, we try to understand the reasons of these results by analyzing more specifically the case using the medium parameterization.

### 5.5.1 Comparison of the mapping functions

The history of the adaption cost function is depicted in figure (29), for the two adaptions that are successively performed at convergence of the optimization procedure (first one in blue, second one in red). Obviously, the adaption cost function value is zero for the initial shape. Then, this figure shows that each of the two optimization processes increases the irregularity of the deformation field, since the initial values of the adaption cost function are respectively 588 and 371. The first adaption is particularly efficient, since the adaption cost function value is reduced by almost two. However, the gain of regularity during the second adaption is moderate.

The function $s = \phi(\xi)$ which defines the mapping is depicted in figure (30), when adaption is performed at convergence. The functions obtained after the first and the second adaption are compared. As can be seen, the mapping change during the second adaption is very small. However, one can notice that the cost function decrease resulting from the second adaption is not negligible (figure (18)). Therefore, it indicates that the results are highly sensitive to the mapping changes.

The first and the last mapping functions when adaption is performed every 10 iterations are depicted in figure (31). It shows that the first adaption, which occurs at the tenth iteration, produces a mapping very close to the final one. It justifies the use of an early adaption and explains why this strategy is very efficient. Indeed, there is no need to achieve the full convergence of the optimization to determine a suitable adapted parameterization.

The mapping functions obtained from the two strategies (adaption at convergence or every 10 iterations) look similar but exhibit some discrepancies (figure (32)) that explain the difference between the results.

The evolution of the cost function related to that of the adaption cost function can be seen in figure (33), for adaption at convergence and adaption every 10 iterations. On this figure, horizontal lines correspond to the adaption procedure (the cost function remains unaltered), whereas diagonal lines correspond to the optimization procedure (both functions vary). The numbers beside the lines precise the iteration numbers for which adaption is carried out. One can notice that the loss of regularity during the first optimization (red dashed line) is almost obtained during the first ten iterations (blue plain line). Moreover, the ratio of the cost function decrease and the adaption cost function increase is maintained until convergence. The adaption cost function reductions during the first adaption are similar for both stretegies. However, the loss of regularity is then lower when adaption is performed progressively (blue line). On the contrary, during the second optimization performed until convergence (red dashed line), the ratio of evolution is the same as that during the first optimization. Finally, the best strategy is the one that maintains the adaption cost function at the lowest value. It is also interesting to observe that the more adaption steps are performed, the less adaption cost function is reduced. But at the same time, the cost function reduction goes on. As already mentioned, this indicates that the cost function is highly sensitive, perhaps more and more sensitive, to the parameterization. Then, one can ask if the ideal parameterization corresponds to the maximum of the sensitivity of the cost function with respect to the mapping, at the optimum design found.

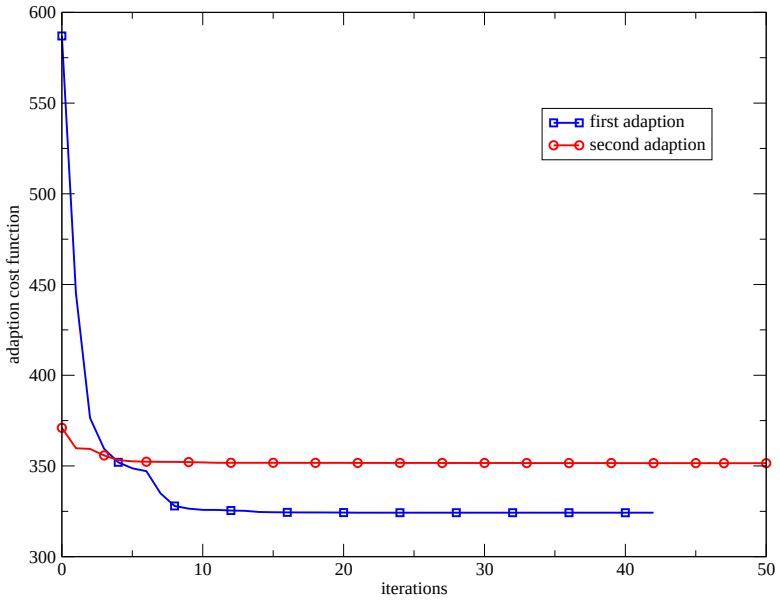



Figure 29: History of the adaption cost function for adaption at convergence.

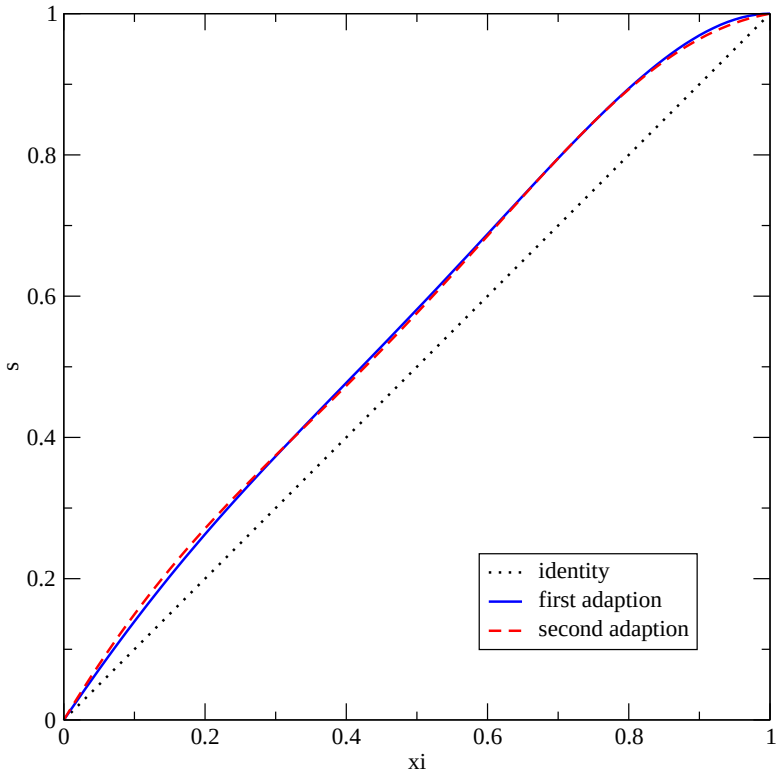



Figure 30: Mapping functions $s = \phi(\xi)$ for adaption at convergence.

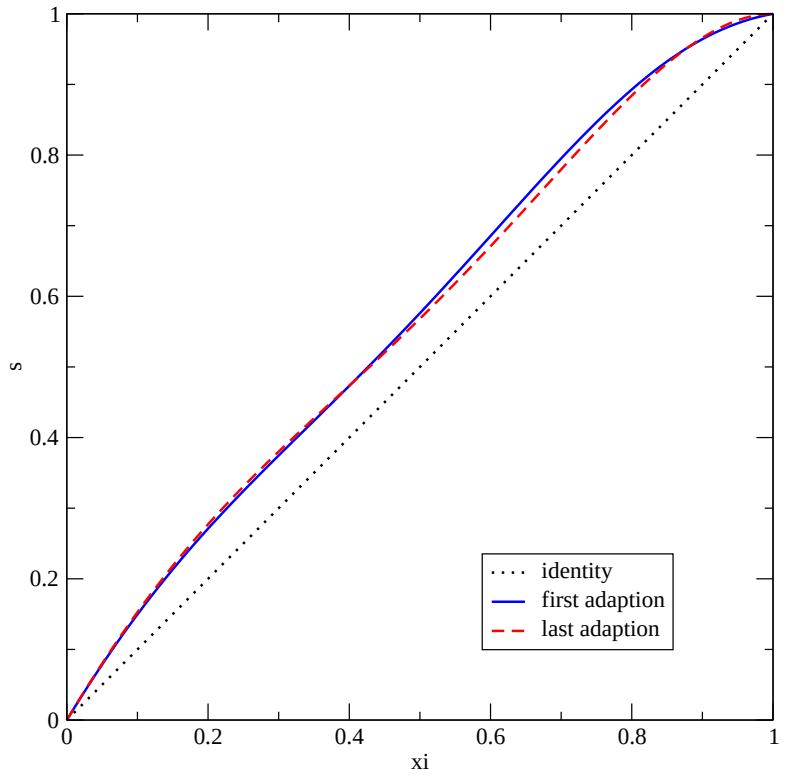



Figure 31: Mapping functions $s = \phi(\xi)$ for adaption every 10 iterations.

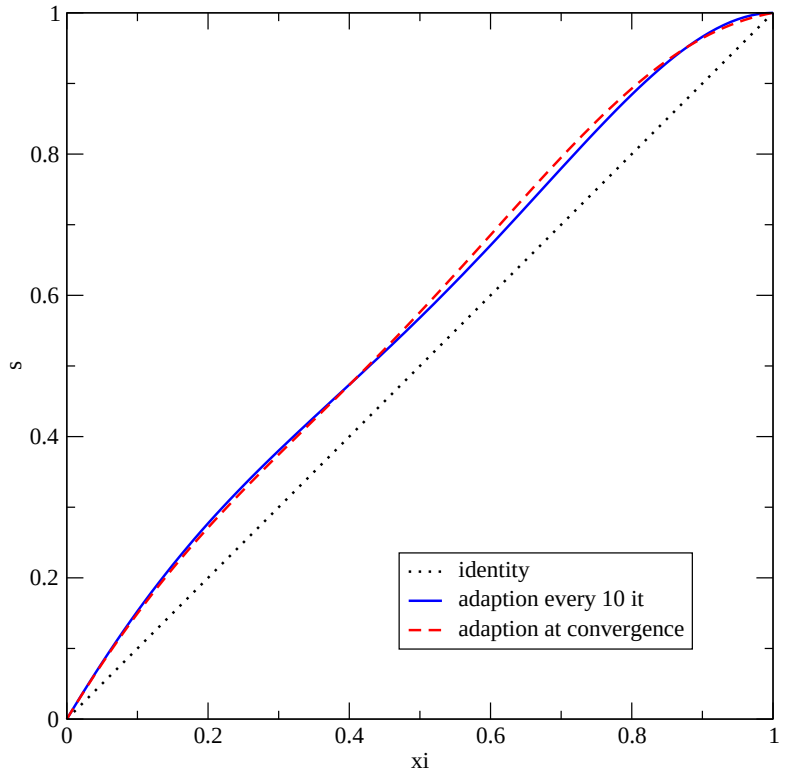



Figure 32: Mapping functions $s = \phi(\xi)$ for adaption at convergence and adaption every 10 iterations.

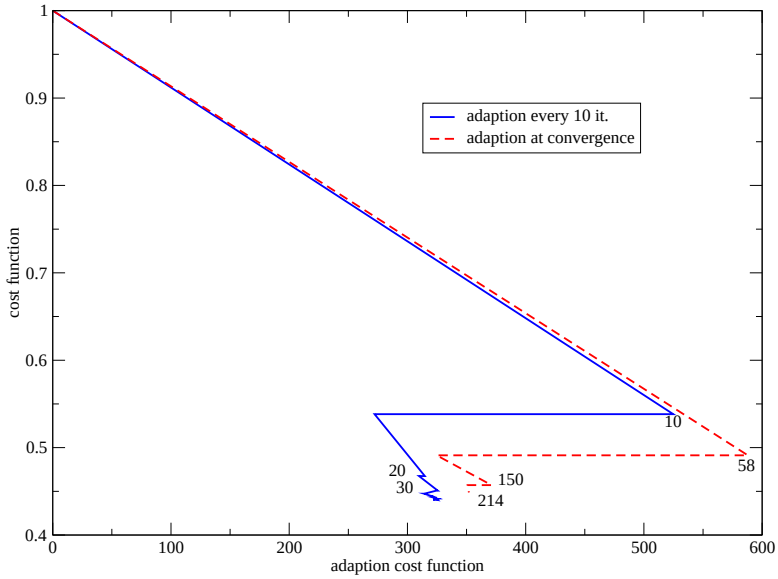



Figure 33: Evolution of the cost function and the adaption cost function.

### 5.5.2 Comparison of the shapes

The optimal shapes found using the initial parameterization, the parameterizations adapted at convergence and adapted every 10 iterations are depicted for some sections in figure (34). It is shown that the discrepancies between the shapes are small, which means that adaption is used by the optimizer to perform some small corrections on the shape. However, these corrections have a significant influence on the fitness, as observed in the cost function decrease (table (1)).

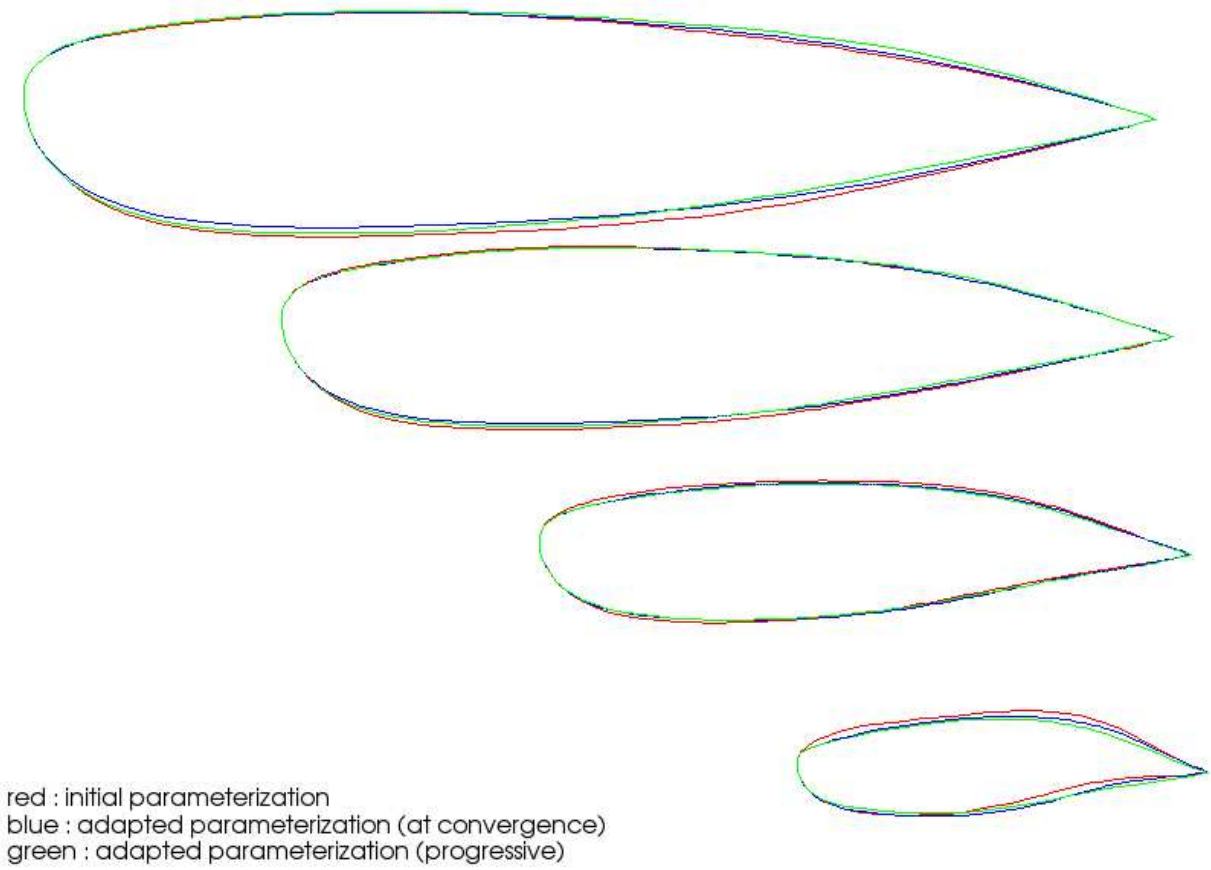



Figure 34: Comparison of the shapes (vertical scale 2) for some sections, for the initial parameterization (red), the parameterization adapted at convergence (blue) and the parameterization adapted progressively (green).

### 5.5.3 Comparison of the flows

As expected, the flow characteristics are highly sensitive to the shape changes. The Mach number fields for the optimal shapes found using the initial parameterization, the parameterizations adapted at convergence and adapted every 10 iterations are compared in figures (35) to (37) for the root section and (38) to (40) for the tip section. Figures (41) and (42) represent the pressure coefficients for the same configurations. Although the shape changes are small, one can see that parameterization adaption yields shapes characterized by a shock wave of lower intensity and of different location, resulting in a better fitness.

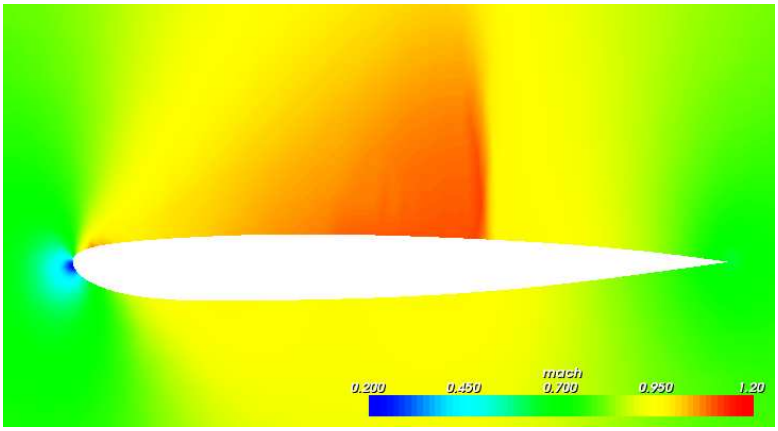



Figure 35: Comparison of the Mach number fields at the root section, for the initial parameterization.

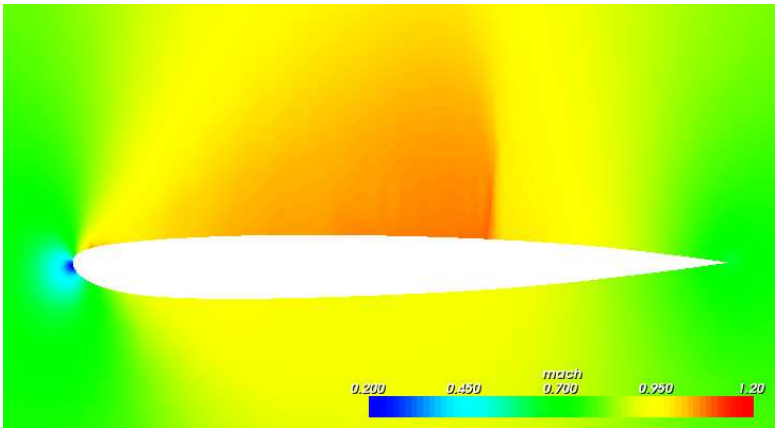



Figure 36: Comparison of the Mach number fields at the root section, for the parameterization adapted at convergence.

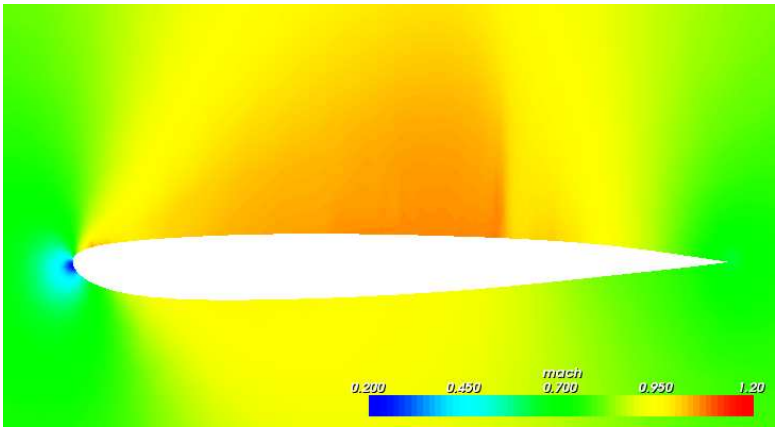



Figure 37: Comparison of the Mach number fields at the root section, for the parameterization adapted progressively.

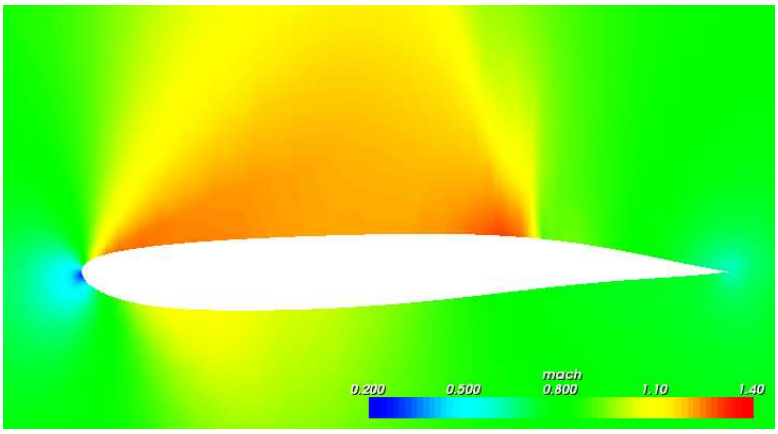



Figure 38: Comparison of the Mach number fields at a section close to the tip, for the initial parameterization.

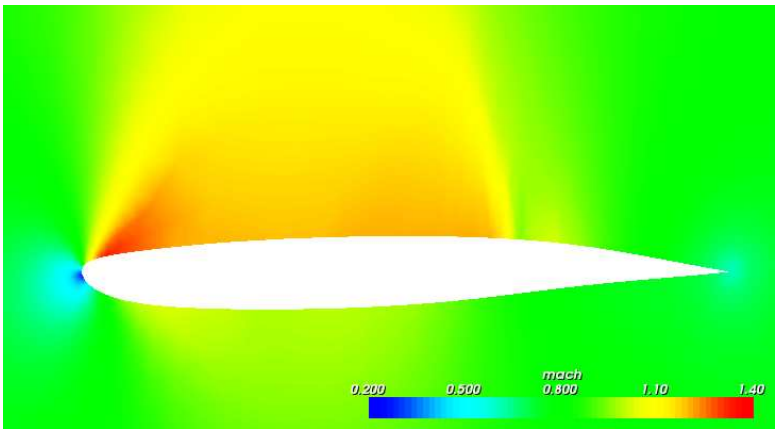



Figure 39: Comparison of the Mach number fields at a section close to the tip, for the parameterization adapted at convergence.

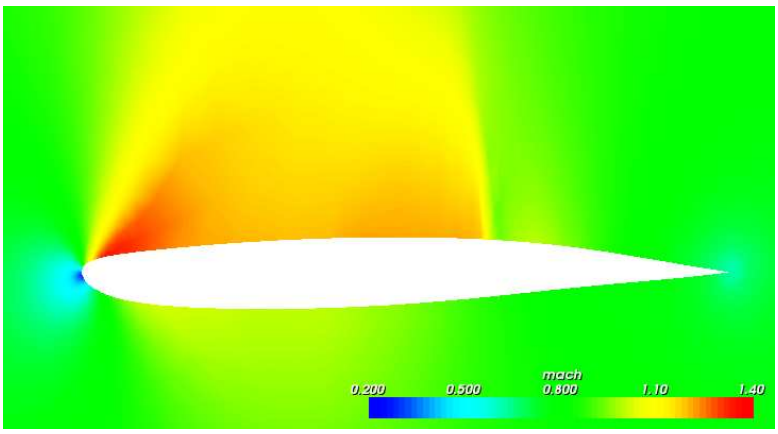



Figure 40: Comparison of the Mach number fields at a section close to the tip, for the parameterization adapted progressively.

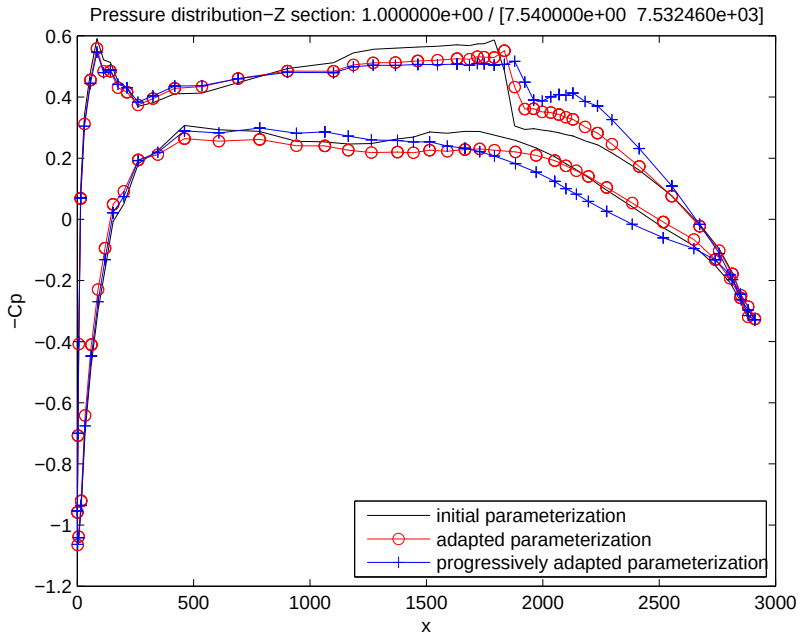



Figure 41: Comparison of the pressure coefficients at the root section, for the initial parameterization (black, no markers), the parameterization adapted at convergence (red, circle markers) and the parameterization adapted progressively (blue,cross markers).

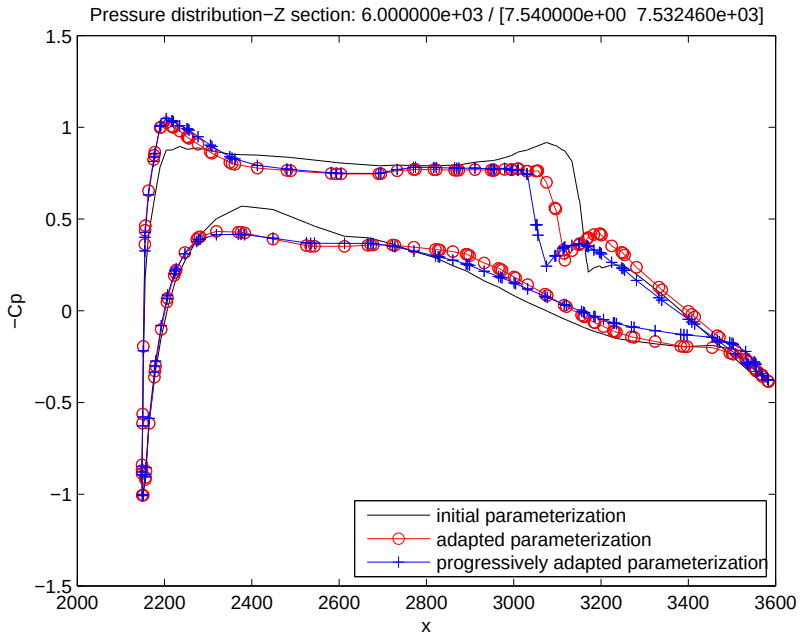



Figure 42: Comparison of the pressure coefficients at a section close to the tip, for the initial parameterization (black, no markers), the parameterization adapted at convergence (red, circle markers) and the parameterization adapted progressively (blue,cross markers).

# Conclusion

A parameterization adaption procedure was developed in the framework of the Free-Form Deformation (FFD) approach. It is based on the optimization of the mapping that defines the FFD coordinates from the lattice coordinates, in order to regularize the displacement of the control points, according to a first approximation of the optimal shape. This approach was tested on the optimization of the wing shape of a business aircraft. It was shown that parameterization adaption permits to reach shapes of better fitness and also to accelerate the convergence. Especially, it was found that the use of a parameterization of lower degree but adapted yields better results than the use of a parameterization of higher degree but naïve.

This study puts in light the critical influence of the parameterization on the results for practical aerodynamic shape optimization problems. Although parametric shape optimization is today a common exercise, the delicate issue of the choice of the parameterization and its influence on the results is often overlooked. This choice has to be made *a priori* although it requires a specific knowledge. Therefore, adaptive algorithms seem well suited to overcome this difficulty. The adaptive parameterization procedure proposed in this report exhibited some promising results. However, several points can still be improved or should be studied :

- during the adaption procedure, the shape changes due to the least-squares approximation of the current shape should be controlled in order to prevent the violation of a constraint ;
- a more general mapping definition, that allows cross-dependencies between the coordinates should be tested ;
- the possibility to replace the mapping as adaption variable by an other monitor should be studied ;
- a theory that justifies precisely and rigorously the choice of the regularization of the deformation field as adaption criterion should be established.

Finally, it would be interesting to evaluate such an adaption parameterization strategy for more complex shape optimization problems, for which the geometrical shape characteristics are more general.

# References

[1] Andreoli, M., Janka, A., and Désidéri, J.-A. Free-form deformation parameterization for multilevel 3D shape optimization in aerodynamics. INRIA Research Report 5019, November 2003.

[2] Clarich, A., and Désidéri, J.-A. Self-adaptative parameterisation for aerodynamic optimum-shape design. INRIA Research Report 4428, March 2002.

[3] Dennis, J., and Torczon, V. Direct search methods on parallel machines. *SIAM Journal of Optimization 1*, 4 (1991), 448–474.

[4] Dervieux, A., and Désidéri, J.-A. Compressible flow solvers using unstructured grids. INRIA Research Report 1732, June 1992.

[5] Désidéri, J.-A. Two-level ideal algorithm for parametric shape optimization. *Journal of Numerical Mathematics 9* (2006).

[6] Désidéri, J.-A., Majd, B. A. E., and Janka, A. Nested and self-adaptive bézier parameterization for shape optimization. In *International Conference on Control, Partial Differential Equations and Scientific Computing, Beijing, China* (September 13-16 2004).

[7] Désidéri, J.-A., and Zolésio, J.-P. Inverse shape optimization problems and application to airfoils. *Control and Cybernatics 34*, 1 (2005).

[8] Farin, G. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, 1989.

[9] Majd, B. A. E., Désidéri, J.-A., and Habbal, A. A fully multilevel and adaptive algorithm for parameterized shape optimization. In *European Multigrid Conference, EMG 2005, Scheveningen, The Netherlands* (September 27-30 2005).

[10] Samareh, J. A survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal 39*, 5 (2001), 877–884.

[11] Samareh, J. Geometry and grid/mesh generation issues for cfd and csm shape optimization. *Optimization and Engineering 5*, 1 (2005), 21–32.

[12] Sederberg, T., and Parry, S. Free-from deformation of solid geometric models. *Computer Graphics 20*, 4 (1986), 151–160.

[13] Torczon, V. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. PhD thesis, Houston, TX, USA, 1989.

[14] Venter, G., and Sobieszczanski-Sobieski, J. Particle swarm optimization. *AIAA Journal 41*, 8 (August 2003), 1583–1589.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399